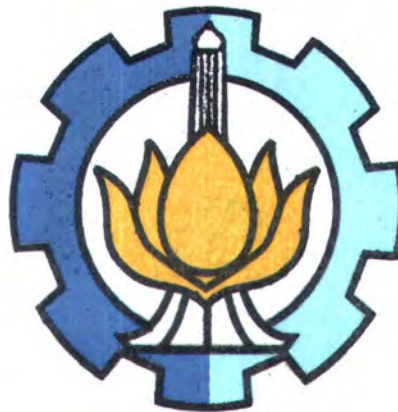


**PEMBUATAN ALAT BANTU PERANCANGAN
DENGAN HIERARCHICAL TASK ANALYSIS
UNTUK REPORT DESIGNER**

TUGAS AKHIR



RSIf
004
Pur
P

2004

Disusun Oleh :

Aryo Purboyo
NRP. 5199.100.013

PERPUSTAKAAN ITS	
Tgl. Terima	11-8-2004
Terima Dari	HI
No. Agenda Prp.	220775

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2004**

PEMBUATAN ALAT BANTU PERANCANGAN DENGAN HIERARCHICAL TASK ANALYSIS UNTUK REPORT DESIGNER

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya


Mengetahui / Menyetujui

Dosen Pembimbing I



Febrilivan S. S.Kom, M.Kom
NIP. 132 206 858

Dosen Pembimbing II



Dwi Sunaryono, S.Kom
NIP. 132 163 671

**SURABAYA
AGUSTUS 2004**

ABSTRAK

Kebutuhan akan informasi yang cepat dan tepat sangat diperlukan sekarang ini dan salah satu faktor penentu kecepatan dalam mendapatkan informasi adalah proses pembuatan sistem informasi. Hal ini dikarenakan kecepatan pembuatan sistem informasi juga akan menentukan kecepatan dalam mendapatkan informasi yang diinginkan. *Hierarchical Task Analysis* (HTA). HTA merupakan suatu metode analisis sederhana yang menggambarkan pemecahan tugas secara bertingkat-tingkat (hirarkis) dan dapat dibagi-bagi menjadi beberapa sub tugas. Pada perkembangan dunia informasi saat ini, banyak ditemui permasalahan dalam hal mendesain aplikasi sistem informasi khususnya desain report. Dengan menggunakan metode HTA tugas-tugas tersebut akan dibagi menjadi beberapa tingkatan dan terbagi lagi menjadi sub-sub tingkatan. *Report Designer* pada umumnya digunakan untuk membuat suatu desain report yang baik. *Report designer* biasanya digunakan oleh pihak pengembang software dalam membuat suatu desain *report* yang baik dan tepat.

Metodologi yang digunakan dalam pembuatan tugas akhir ini terdiri dari beberapa tahapan. Pertama-tama dilakukan studi literatur untuk mempelajari teori HTA serta aplikasi *report designer* yang sudah ada. Setelah itu dilakukan pemodelan permasalahan dalam pembuatan *report designer* dengan menggunakan metode HTA. Kemudian dilakukan uji coba dan evaluasi terhadap sistem yang telah dibuat. Terakhir, dilakukan penyusunan dokumentasi dalam bentuk buku tugas akhir.

Uji coba penggunaan aplikasi dilakukan untuk mengetahui sejauh mana keberhasilan sistem yang telah dikembangkan. Parameter yang digunakan dalam aplikasi ini adalah kecocokan dan ketepatan pencetakan sesuai dengan desain *report*. Sebagai hasil akhir dari sistem HTA ini, para pengguna akan mendapatkan kemudahan dalam membuat aplikasi *report designer*.

Kata Kunci : *Hierarchical Task Analysis, Report Designer*

KATA PENGANTAR

Alhamdulillah *rabbil 'alamin*, segala puji bagi Allah Yang Maha Kuasa yang telah memberikan kekuatan-Nya sehingga penulis bisa menyelesaikan tugas akhir yang berjudul : **“PEMBUATAN ALAT BANTU PERANCANGAN DENGAN HIERARCHICAL TASK ANALYSIS UNTUK REPORT DESIGNER”**.

Melalui tugas akhir ini, penulis merasa mendapat kesempatan besar untuk lebih memperdalam ilmu pengetahuan yang diperoleh selama di perkuliahan, terutama berkenaan dengan implementasi teknologi informasi dalam kehidupan sehari-hari. Namun demikian, penulis menyadari bahwa tugas akhir ini masing memiliki banyak kelemahan dan kekurangan. Oleh karena itu penulis sangat mengharapkan kritik dan sarannya bagi pengembangan ke depannya.

Akhirul kalam, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penyelesaian tugas akhir ini. Semoga Allah membalasnya dengan balasan sebaik-baiknya. Amien.

Surabaya, Agustus 2004

Penulis

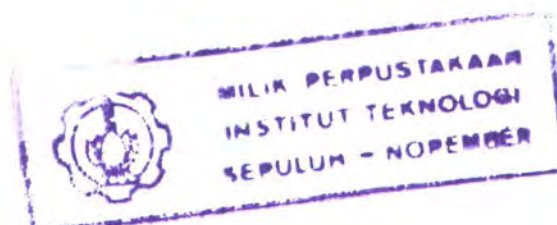
UCAPAN TERIMA KASIH

Syukur Alhamdulillah selalu terpanjat kepada Allah Subhana wa Ta'ala yang telah memberikan rahmat dan hidayah serta banyak kemudahan selama mengerjakan tugas akhir ini.

Dalam kesempatan ini saya mengucapkan terima kasih dan penghargaan sebesar-besarnya bagi semua pihak yang telah membantu dalam pelaksanaan Tugas Akhir ini, baik secara langsung maupun tidak langsung. Secara khusus saya mengucapkan terima kasih kepada :

1. Ibu dan Ayah penulis yang memberikan perhatian, nasehat, bantuan material dan spiritual, serta doa yang tak ternilai besarnya demi kelancaran tugas akhir ini.
2. Bapak Febriliyan Samopa S.Kom, M.Kom dan Bapak Dwi Sunaryono S.Kom selaku Dosen Pembimbing Tugas Akhir yang senantiasa memberikan bimbingan dan petunjuk.
3. Bapak Ir. Yudhi Purwananto M.Kom selaku Ketua Jurusan Teknik Informatika ITS.
4. Bapak Prof. DR. Ir. Arif Djunaidy M.Sc, Ph.D selaku Dosen Wali penulis yang telah membimbing penulis selama kuliah.
5. Bapak dan Ibu dosen yang telah membimbing dan membagikan ilmunya sehingga penulis layak mendapatkan gelar sarjana.
6. Segenap kerabatku tercinta: Bu Nanik dan Angga semoga diberi ketabahan, Pakde dan Bude Yudo serta buat Mbak diyah, Mas Satriyo, Diyon dan Pandu.

- Ibu Jagir, Mas Edi, Mas Budi, dan seluruh sepupu dan keponakan. Terima kasih atas kebersamaannya selama ini.
7. Adik Keke tersayang yang telah memberi semangat, dorongan serta berbagi suka dan duka selama kuliah di TC.
 8. Nafis, Ario, Hendra, Maman, Rachmad, Kamal, Medi, Anna serta Keluarga Besar angkatan 99/COF dengan kebersamaan yang terbina selama ini.
 9. Aniz Ext'00, atas pinjaman printer buat mengerjakan dan sidang tugas akhir ini.
 10. Rekan Laboratorium Sistem Informasi (Budi, Warna, Shidiq, Ridha) yang banyak membantu dalam mengerjakan tugas akhir.
 11. Audrey, Intan, Pak Ivan, Mas Gun, Mas Toyyib dan Pak Totok yang telah banyak mengisi hari-hari penulis dengan canda dan tawa.
 12. Seluruh rekan yang hadir pada waktu sidang tugas akhir.
 13. Seluruh Staf dan Karyawan Tata Usaha Jurusan Teknik Informatika ITS yang telah membantu kemudahan administrasi dan fasilitas.
 14. Semua pihak yang telah membantu dalam penyelesaian tugas akhir ini.



DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR.....	ii
UCAPAN TERIMA KASIH.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.2 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Metodologi Pembuatan Tugas Akhir	4
1.6 Sistematika Pembahasan	6
BAB II DASAR TEORI.....	8
2.1 Hierarchical Task Analysis (HTA)	9
2.1.1 Perbedaan Hierarchical Task Analysis (HTA) dan Procedural Task Analysis.....	14
2.1.2 Konsep Umum Hirarki (Tingkatan)	17
2.1.3 Konsep Umum Analisis Tugas.....	19
2.1.4 Contoh Penggunaan Hierarchical Task Analysis (HTA) pada Pembuatan Reading Web Site	21

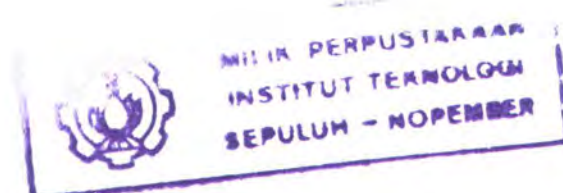
2.2	Hubungan Hierarchical Task Analysis (HTA) dengan Report Designer	25
2.2	SQL Server.....	26
2.3	Microsoft Access.....	28
2.4	Visual Basic.....	29
BAB III PERANCANGAN SISTEM		31
3.1	Deskripsi Umum Sistem.....	31
3.2	Spesifikasi Kebutuhan Sistem.....	33
3.2.1	Pengguna Aplikasi.....	33
3.2.2	Masukan dan Keluaran Sistem.....	33
3.2.3	Hubungan Hirarchical Task Analysis (HTA), Report Designer dan Aplikasi Sistem	34
3.3	Perancangan Proses.....	35
3.3.1	DFD Level 0.....	35
3.3.2	DFD Level 1.....	37
3.3.3	DFD Level 2.....	38
3.3.4	DFD Level 3.....	42
3.4	Perancangan Data.....	47
3.4.1	Model Data Konseptual.....	48
3.5	Perancangan Antarmuka	49
BAB IV IMPLEMENTASI PERANGKAT LUNAK		53
4.1	Lingkungan Implementasi.....	53
4.2	Implementasi Data.....	54
4.2.1	Model Data Fisik.....	54

4.3	Implementasi Proses.....	58
4.3.1	Penentuan dan Perancangan Desain <i>Report</i>	58
4.4	Implementasi Antarmuka	62
4.4.1	Antarmuka Design Report.....	62
1.4.2	Antarmuka Pemilihan Kode Akses	63
1.4.3	Antarmuka Obyek Visual Basic yang Digunakan.....	64
1.4.4	Antarmuka Properti dari Obyek yang Digunakan.....	65
1.4.5	Antarmuka Fungsi	66
1.4.6	Antarmuka Pengaturan Kertas dan Margin.....	67
BAB V	UJI COBA.....	69
5.1	Lingkungan Pengujian Aplikasi	69
5.2	Pelaksanaan Uji Coba.....	69
BAB VI	PENUTUP	88
6.1	Kesimpulan.....	88
6.2	Kemungkinan Pengembangan.....	89
DAFTAR PUSTAKA		90

DAFTAR GAMBAR

Gambar 2.1	HTA : Tugas <i>Housekeeping</i>	10
Gambar 2.2	HTA : Making a cup of tea	11
Gambar 2.3	<i>Modified task hierarchy for making lots of cups of tea</i>	11
Gambar 2.4	Konsep Hierarki	17
Gambar 2.5	Contoh HTA :oleh Komar Khan from EDCI 705, Fall 1999	21
Gambar 2.6	Obyek Basis Data Access	28
Gambar 3.1	Arsitektur Sistem	34
Gambar 3.2	DFD Level 0	36
Gambar 3.3	DFD Level 1	37
Gambar 3.4	DFD Level 2 – Menentukan Tujuan Sistem Aplikasi	39
Gambar 3.5	DFD Level 2 – Melakukan analisis tugas	40
Gambar 3.6	DFD Level 2 –Menentukan Tindakan yang Diambil	40
Gambar 3.7	DFD Level 2 –Melakukan Tindakan	41
Gambar 3.8	DFD Level 3 – Penentuan dan Perancangan Desain Aplikasi	43
Gambar 3.9	DFD Level 3 – Penentuan dan Perancangan Desain Report	44
Gambar 3.10	DFD Level 3 – Menentukan Obyek	45
Gambar 3.11	DFD Level 3 – Menentukan Event	46
Gambar 3.12	DFD Level 3 – Desain Antarmuka Form	47
Gambar 3.13	Diagram Data Konseptual – Sistem HTA	48
Gambar 3.14	Antarmuka Komponen yang digunakan ..	49
Gambar 3.15	Antarmuka <i>Property</i> dari Komponen	50

Gambar 3.16	Antarmuka Kode Akses Form	51
Gambar 3.17	Antarmuka Fungsi yang Digunakan	51
Gambar 3.18	Antarmuka Pengaturan Kertas dan Margin	52
Gambar 4.1	Diagram Data Fisik – Sistem HTA	54
Gambar 4.2	Pseudocode untuk Mengambil Kode Akses	59
Gambar 4.3	Pseudocode untuk Menampilkan Desain <i>Report</i>	60
Gambar 4.4	Pseudocode untuk Menyimpan Desain <i>Report</i>	60
Gambar 4.5	Subrutin untuk Mengurutkan Obyek	61
Gambar 4.6	Antarmuka Desain Report	62
Gambar 4.7	Antarmuka Pemilihan Kode Akses	63
Gambar 4.8	Antarmuka Obyek VB	64
Gambar 4.9	Antarmuka property dari VB	66
Gambar 4.10	Antarmuka Fungsi	66
Gambar 4.11	Antarmuka Pengaturan Kertas dan Margin	68
Gambar 5.1	Uji Coba Desain <i>Report</i>	71
Gambar 5.2	Hasil Uji Coba Skenario 1	72
Gambar 5.3	Uji Coba Desain Report dengan <i>Field</i> hasil kueri	73
Gambar 5.4	Hasil Pencetakan Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i>	74
Gambar 5.5	Uji Coba Desain Report dengan <i>field</i> hasil <i>query</i> dan Fungsi	76
Gambar 5.6	Hasil Pencetakan Uji Coba Desain Report dengan <i>Field</i> hasil <i>Query</i> dan Fungsi	77



Gambar 5.7 Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil	79
Gambar 5.8 Hasil Pencetakan Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil	80
Gambar 5.9 Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil bersebelahan dengan header.....	81
Gambar 5.10 Hasil Pencetakan Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil bersebelahan dengan header.....	82
Gambar 5.11 Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil beserta footer	84
Gambar 5.12 Hasil Pencetakan Uji Coba Desain Report dengan <i>Field</i> hasil <i>query</i> dan data detil beserta footer	85

DAFTAR TABEL

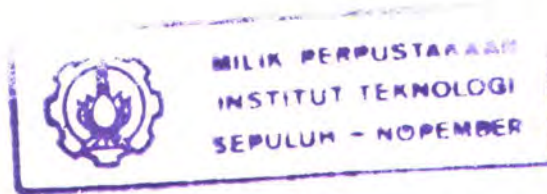
Tabel 2.1	Perbedaan HTA dan Procedural Task Analysis	14
Tabel 3.1	Definisi Entitas DFD Level 0	34
Tabel 4.1	ListCommandSQL	56
Tabel 4.2	ContentPrint	57
Tabel 4.3	Fungsi	58
Tabel 5.1	Lingkungan Pengujian Aplikasi	69

BAB I

PENDAHULUAN

Pada Bab Pendahuluan ini akan dijelaskan hal-hal yang melatarbelakangi pembuatan tugas akhir ini, tujuan, permasalahan yang dihadapi, batasan masalah dan metodologi pembuatannya.

1.1 Latar Belakang



Hierarchical Task Analysis (HTA) merupakan salah satu topik penting pada dunia Interaksi Manusia dan Komputer. HTA merupakan suatu metode analisis sederhana yang menggambarkan pemecahan tugas secara bertingkat-tingkat (hirarkis) dan dapat dibagi-bagi menjadi beberapa sub tugas [1]. Tugas sendiri merupakan rangkaian tindakan atau aktivitas yang diperlukan untuk mencapai suatu tujuan [1]. Usaha penyelesaian tugas terdiri dari dekomposisi tugas dan urutan kelakuannya. Hasil dari HTA berupa sekumpulan tugas dan subtugas yang tersusun secara hirarki dan perencanaan/strategi yang melukiskan semua kondisi dari pekerjaan tugasnya [1].

Biasanya seorang *programmer* akan membuat program melalui form dan kemudian dia menggambar form dengan berbagai macam kontrol yang tersedia. Dalam pustaka Visual Basic, kontrol standar terdapat pada pustaka VB, VBA, VBRUN, sebagai contoh control *command button* terdapat pada pustaka VB atau dikenal sebagai *class member of VB*. Setelah obyek diletakkan pada form, maka secara otomatis tersedia pula *property*, *method* dan *event*. Seorang programmer dapat melakukan setting pada *property*, misalnya Caption =

“&Update”, hal ini bisa dilakukan pada pada *design-time* maupun *run-time*. Pada kasus *design-time*, obyek memiliki keterbatasan terhadap perubahan yang sifatnya statis dan akan berkembang sesuai dengan banyaknya kasus yang ditangani. Sebagai contoh pada kasus basis data transaksional akademik sederhana terdapat sejumlah tabel.

Tabel master :

- Mahasiswa
- Matakuliah
- Dosen

Tabel Transaksional :

- JadwalPelajaran
- PesertaKuliah

Terdapat 5 tabel yang secara spesifik melakukan operasi yang berbeda. Pada pendekatan *design-time* minimal diperlukan 5 interaksi yang bisa terdiri dari beberapa form atau beberapa sstab (salah satu komponen dalam visual basic 6) dan masing-masing interaksi terdapat kontrol obyek dengan sifat yang mungkin berbeda satu sama lain. Jumlah interaksi, obyek dan kode operasi akan semakin besar seiring bertambahnya kasus yang diselesaikan.

Pendekatan *design-time* lebih menonjolkan aspek visualisasi obyek, tanpa melihat optimasi program dimana terdapat perulangan yang sama baik pada kode program ataupun obyek yang dibuat [5]. Seperti kasus sederhana di atas, adalah sejumlah form yang sama akan di tambahkan (*add*). Pertambahan obyek yang sama akan membuat hasil kompilasi (file *.exe) menjadi besar. Pada sebagian kasus contoh lengkap penyelesaian kasus dengan pendekatan *design time* rata-

rata hampir 500 Kbytes. Kelemahan ini dapat di atasi dengan pembuatan obyek sebagai instan dari kelas.

Report merupakan ringkasan dari data yang diolah oleh sistem informasi dan dilaporkan kepada pengguna dan dibuat berdasarkan kebutuhan pengguna [1]. *Report Designer* pada umumnya digunakan untuk membuat suatu desain report yang baik. *Report designer* biasanya digunakan oleh pihak pengembang software dalam membuat suatu desain *report* yang baik dan tepat. Dalam proses HTA ini *report designer* digunakan untuk menunjang kinerja dari aplikasi pembuatan dan perancangan HTA itu sendiri, aplikasi *report designer* ini hanya digunakan untuk merancang desain laporan dari sebuah aplikasi sistem informasi dimana nanti hasil dari *report designer* disimpan dalam sebuah basis data HTA, yang kemudian basis data tersebut akan dibaca oleh sebuah aplikasi sistem informasi yang di buat dengan menggunakan metode HTA.

1.2 Perumusan Masalah

Tugas akhir ini akan membahas bagaimana membuat suatu alat bantu perancangan *hierarchical task analysis* khususnya *report designer* yang mampu menggambarkan obyek yang umum digunakan dalam Visual Basic 6.

1.2 Batasan Masalah

Pada tugas akhir ini perlu didefinisikan beberapa batasan mengenai sejauh mana tugas akhir ini akan dikerjakan. Beberapa batasan masalah tugas akhir ini antara lain:

1. Alat bantu perancangan ini akan merumuskan properti yang umum digunakan pada obyek-obyek Visual Basic 6

2. Pembuatan alat bantu perancangan HTA ini terbatas pada pembuatan *report designer* saja.
3. *Report designer* ini hanya bisa mengimplementasikan fungsi–fungsi yang telah ada dan dipakai dalam aplikasi yang dibangun, aplikasi *report designer* tidak dapat membuat fungsi–fungsi baru.
4. Obyek visual basic yang digunakan dalam aplikasi ini adalah obyek yang umum digunakan dalam desain report.

1.4 Tujuan

Tujuan akhir dari tugas akhir ini antara lain untuk membuat suatu alat bantu perancangan HTA khususnya *report designer*. Alat ini akan merumuskan properti yang umum digunakan pada obyek-obyek Visual Basic 6. Dengan bantuan aplikasi *report designer* ini, diharapkan pengguna merasa terbantu untuk membuat suatu desain *report* yang baik.

1.5 Metodologi Pembuatan Tugas Akhir

Pembuatan tugas akhir ini dilakukan dengan mengikuti metodologi sebagai berikut:

1. Studi literatur dan software yang ada

Pada tahap ini akan dipelajari sejumlah literatur mengenai konsep dan teknologi yang akan digunakan. Literatur yang digunakan meliputi buku referensi, majalah, dokumentasi internet, dan tugas akhir mahasiswa lain yang berhubungan dengan proses bisnis sistem pembayaran atau perangkat lunak pada perangkat bergerak serta tidak lupa pula masukan-masukan yang diberikan.

2. Perancangan perangkat lunak

Pada tahap ini akan dibuat alur proses bisnis yang terjadi serta melakukan pemodelan sistem menggunakan diagram aliran data / *Data Flow Diagram*.

3. Pembuatan Perangkat Lunak

Pada tahap ini, model dan rancangan sistem yang telah dibuat akan diimplementasikan. Perangkat lunak yang akan digunakan dalam implementasi ini adalah Visual Basic 6, SQL Server dan basis data MS Access.

4. Uji Coba dan Evaluasi

Pada tahap ini, asumsi implementasi sudah selesai, dilakukan uji coba kebenaran dengan beberapa data yang telah disiapkan. Kemudian, hasil pengujian ini akan dievaluasi untuk menentukan validitas sistem yang dibuat sebagai bahan pertimbangan perlu tidaknya melakukan perbaikan pada program.

5. Penyusunan Buku Tugas Akhir

Tahap ini merupakan tahap terakhir dari proses pengerjaan Tugas Akhir ini. Buku dokumentasi akan disusun sebagai laporan dari seluruh proses pengerjaan Tugas Akhir ini. Diharapkan dari penyusunan buku Tugas Akhir ini akan dapat menjadi referensi bagi orang lain yang ingin melakukan pengembangan sistem lebih lanjut.

1.6 Sistematika Pembahasan

Buku Tugas Akhir ini terdiri atas beberapa bab sebagai berikut:

Bab I, Pendahuluan

Bab ini menjelaskan tentang latar belakang yang mendasari dikerjakannya tugas akhir ini, perumusan masalah, batasan masalah, tujuan, dan metodologi pembuatan Tugas Akhir.

Bab II, Teori Penunjang

Bab ini menjelaskan secara singkat tentang teori-teori yang digunakan selama proses pengerjaan tugas akhir ini, antara lain: Pengertian HTA beserta contohnya serta penggunaannya dalam desain, SQL Server, Visual Basic dan Microsoft Access.

Bab III, Perancangan

Bab ini menjelaskan tentang perancangan perangkat lunak dari sistem yang akan dibuat, meliputi: Deskripsi Umum Sistem, Arsitektur Sistem, Data Flow Diagram, Desain Antar Muka, dan Desain Basis data.

Bab IV, Implementasi

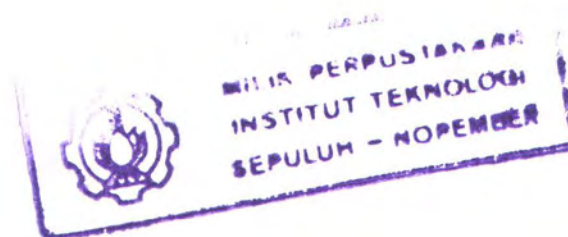
Bab ini berisi tentang pembuatan aplikasi, meliputi bagaimana aplikasi dapat menjelaskan proses pembuatan HTA khususnya pada perancangan *report designer* serta bagaimana mengimplementasikan hasil pembuatan HTA dengan menggunakan *report designer* pada aplikasi sistem informasi.

Bab V, Uji Coba dan Evaluasi

Bab ini menjelaskan tentang proses pengujian terhadap sistem yang telah dibuat dan melakukan evaluasi terhadap hasil pengujian tersebut.

Bab VI, Kesimpulan dan Saran

Bab ini menjelaskan tentang kesimpulan dari sistem yang dibuat serta saran untuk kepentingan pengembangan lebih lanjut.



BAB II

DASAR TEORI

Dalam bab II ini akan dibahas beberapa teori dasar untuk menunjang penyelesaian tugas akhir ini, antara lain mengenai *Hierarchical Task Analysis* (HTA) , SQL Server, Visual Basic, dan Microsoft Access.

Berikut ini penjelasan singkat mengenai *Task Analysis* dan *Task decomposition Task Analysis* merupakan suatu proses menganalisis cara-cara orang dalam melakukan pekerjaannya seperti apa saja yang mereka lakukan, apa saja yang harus mereka patuhi dan apa saja yang perlu mereka ketahui [1]. Misalnya saja jika mengenai pekerjaan seorang *housekeeping* :

Aturan untuk membersihkan rumah:

- Keluarkan vacuum cleaner
- Pasangkan bagian-bagiannya
- Bersihkan ruangan
- Ketika kotak sampahnya penuh, kosongkan kembali
- Letakkan kembali vacuum cleaner dan perkakasnya kembali setelah digunakan

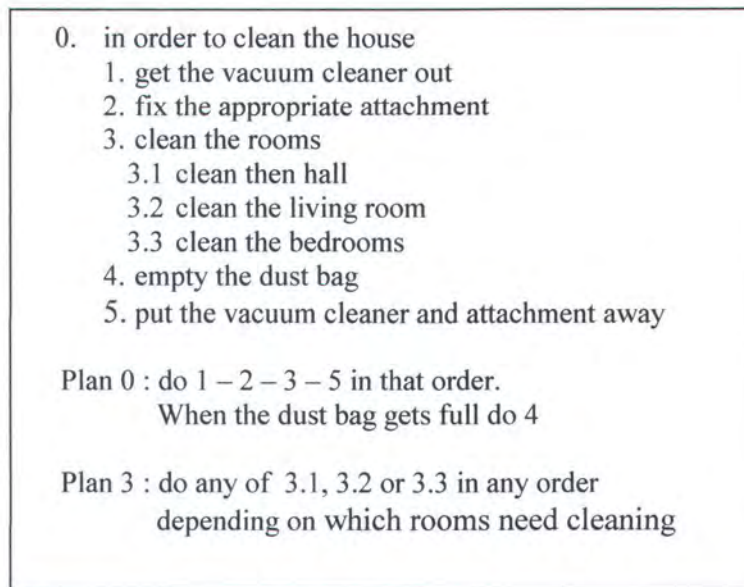
Untuk melaksanakan tugas semacam itu, perlu diketahui tentang vacuum cleaner itu sendiri, peralatannya, kotak debunya, lemari tempat penyimpanannya, ruangan yang akan dibersihkan dan lain sebagainya.

Diperlukan empat pendekatan yang berbeda untuk menganalisis tugas, yang saling tumpang tindih tetapi yang meletakkan tekanannya pada area kecil yang berbeda [1]. Keempat pendekatan tersebut sebagai berikut :

1. **Task Decomposition** merupakan hal yang lebih kepada bagaimana cara membagi suatu tugas menjadi sub-sub tugas.
2. **Knowledge-based Techniques** seperti apa saja kebutuhan user untuk mengetahui tentang obyek dan aksi yang termasuk di dalam tugas, serta bagaimana pengetahuan tersebut diorganisasikan.
3. **Entity-Relation-Based Analysis** merupakan sebuah pendekatan berbasis obyek dimana penekanannya terletak pada pengidentifikasian aktor dan obyek, hubungan/relasi diantaranya dan aksi yang dilakukan.
4. **Task Analysis** lebih mengarah pada bagaimana cara pengadaan sistem dan prosedur, dan peralatan utamanya adalah semua observasi/penelitian dalam berbagai bentuk sistem aplikasi sistem informasi seperti : kepegawaian, akuntansi, gudang, inventori dan lain sebagainya. Salah satu tujuan dari *task analysis* adalah untuk membantu dalam menghasilkan pemecahan masalah pokok/utama dan dokumentasi. Untuk itu, analisis terhadap sistem yang telah ada cukup penting.

2.1 Hierarchical Task Analysis (HTA)

HTA merupakan suatu metode analisis sederhana yang menggambarkan pemecahan tugas secara bertingkat-tingkat (hirarkis) dan dapat dibagi-bagi menjadi beberapa sub tugas [1] . Dan karena suatu tugas dapat diuraikan ke dalam sub-sub tugas atau dipartisi, maka sebuah rancangan tertentu/khusus dapat diberikan pada masing-masing tugas tersebut. Contoh pembagian tugas *housekeeping* dibagi menjadi beberapa sub tugas dapat dilihat pada gambar 2.1 di bawah ini.

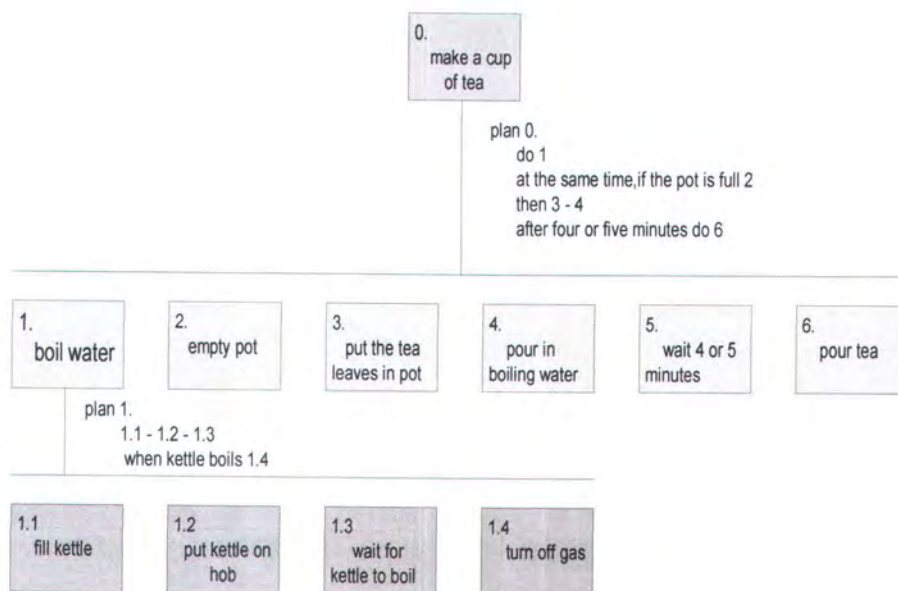


Gambar 2.1 HTA : Tugas Housekeeping

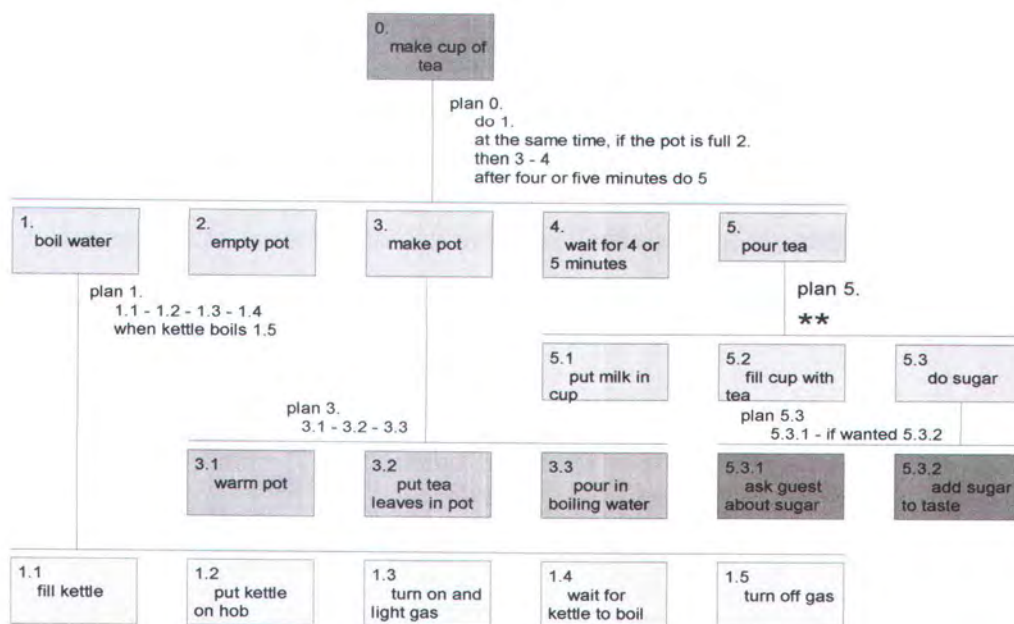
Output dari HTA adalah sebuah hirarki tugas-tugas dan sub-sub tugas serta rencana-rencana yang menggambarkan dalam hal permintaan apa dan kondisi apa sub-sub tugas tersebut dilaksanakan.

Sebuah task analysis juga dapat direpresentasikan dalam bentuk diagram sebaik dalam bentuk teks. Contohnya dalam hal ini tugas utamanya adalah membuat secangkir teh yang didekomposisikan menjadi 6 subtugas yang dapat dilihat pada gambar 2.2

Satu cara bagi seorang analis untuk mencari kelalaian/kesalahan semacam itu adalah dengan cara pengujian pada format subtugas. Contohnya katakanlah “matikan kompor” tetapi tidak pernah disebutkan “nyalakan kompor”. Hal-hal semacam ini membuat hirarki menjadi sedikit tidak imbang. Maka hasil modifikasi HTA setelah semua analisis diberikan dapat dilihat pada gambar 2.3

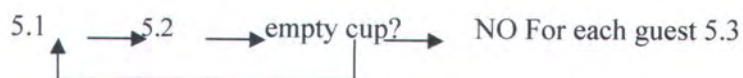


Gambar 2.2 Hierarchical Task Analysis : making a cup of tea



Gambar 2.3 Modified task hierarchy for making lots of cups of tea

Keterangan dari gambar 2.3 :



sebersih atau sekotor apapun yang diinginkan berarti ada hal-hal yang didiskritkan atau dipastikan sesuai aturan tertentu.

7. *Mixture* kebanyakan rencana merupakan campuran dari element-element tersebut. Contohnya, pada plan1 untuk *boiling water* secara luas merupakan urutan yang sudah tetap/pasti tetapi dipisahkan dengan menunggu berarti ada campuran antar tugas yang satu dengan yang lain (tugas memasak air sampai mendidih dan tugas menunggu air mendidih).

HTA dapat digambarkan dalam tiga langkah: *Starting – Progressing – Finalizing* [1].

1. Memulai Analisis (Starting the Analysis)

- a. Tentukan tugas utamanya (*main task*).
- b. Bagi menjadi beberapa sub task. Dalam *report designer* ini tergantung dari banyaknya kode akses yang ada. Sub-sub tugas tersebut harus menggambarkan tujuan-tujuan dan meliputi keseluruhan kebutuhan tugas utama.
- c. Gambarkan sub-sub tugas sebagai layer/lapisan-lapisan. Buatlah rancangan bagaimana sub-sub tugas tersebut saling terhubung.

2. Kemajuan Analisis (Progressing the Analysis)

- a. Tentukan detail-detail pada tiap tingkatan.
- b. Tentukan pada tiap-tiap tugas apakah analisisnya harus dilanjutkan.

3. Penyelesaian Analisis (Finalizing the Analysis)


- a. Periksa setiap hasil dekomposisi semua alternative yang dapat dicakup.
- b. Lakukan evaluasi dan penilaian terhadap hasil dekomposisi dengan ahlinya.

2.1.1 Perbedaan Hierarchical Task Analysis (HTA) dan Procedural Task Analysis

Perbedaan antara *Hierarchical Task Analysis* dengan *Procedural Task Analysis* dapat dilihat pada tabel 2.1 di bawah ini [4].

Tabel 2.1 Perbedaan HTA dan Procedural Task Analysis

Perbandingan	Hierarchical Task Analysis (HTA)	Procedural Task Analysis
Definisi	Merupakan analisis tugas yang berurutan / hirarkis dikenal pula sebagai tugas prasyarat, semisal apa yang harus user ketahui dan apa yang harus dilakukan user untuk mencapai tujuan tugas tersebut.	Analisis tugas mengenai cara atau suatu pengolahan analisis informasi. Semisal apa saja yang menjadi langkah-langkah fisik agar user berhasil dalam rangka melengkapi dan menyelesaikan suatu tugas.
Pengembangan	Dikembangkan dari bawah ke atas (bottom up) atau dari hal-hal yang bersifat umum ke hal-hal yang sifatnya lebih spesifik	Dikembangkan secara linier dan secara sekuen, <i>step by step</i> . Serta mempunyai suatu arus directional yaitu mempunyai suatu start awal dan suatu akhir.
Tingkatan	Seperti pada pelajaran toksonomi, ada tingkatan-	Tidak ada kaitan dengan tingkatan seperti pada pelajaran

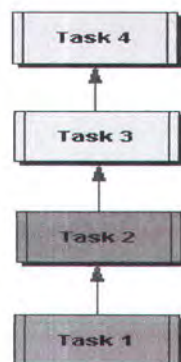
	<p>tingkatan yang harus dilalui, dimulai dari hal-hal yang paling rumit sampai kepada yang paling sedikit kompleksitasnya</p> 	<p>taksonomi, melainkan berkaitan dengan cara secara alamiah. Jadi tugas adalah suatu relasi aturan, kemudian langkah-langkah dari analisis tugas akan meliputi bagaimana cara [menerapkan / memberlakukan] aturan ini. Jika tugas adalah konsep yang mempelajari, maka kemudian analisis tugas akan meliputi " bagaimana cara menentukan apakah kejadian tertentu adalah suatu contoh [dari / tentang] konsep ini".</p>
<p>Bentuk</p> <p>Distribusi</p>	<p>Ada tingkatan tugas. Tingkatan yang paling tinggi menjadi tugas yang paling rumit dan menjadi format prasyarat bagi tingkat yang lebih rendah. Masing-masing tugas dapat dipecah ke dalam satu atau lebih tugas, dan tiap tugas mengacu pada tugas</p>	<p>Diwakili dalam bentuk suatu flowchart atau suatu tatanan garis besar. Jika suatu flowchart yang digunakan, arah dari panah yang menghubungkan antar tugas menandai adanya urutan langkah-langkah tugas. Bentuk intan menjadi lambang keputusan yang menandai adanya suatu perubahan pada arah tergantung</p>

	berikutnya	pada hasilnya. Sedangkan jika suatu garis besar yang digunakan, langkah-langkah di dalam garis besar dinomori untuk menandai adanya urutan tersebut. Subtasks juga dinomori untuk menandai adanya arus di dalam suatu tugas menjadi lebih besar
Pembacaan Analisis	Analisisnya dibaca dari bawah ke atas. Ditunjukkan dengan arah panah yang terus naik menggambarkan hubungan antar tugas ke arah tugas terminal/tujuan.	Dibaca dari kiri ke kanan atau dari atas sampai ke bawah, sesuai arah panah di dalam format flowchart, atau nomor jumlah dari langkah jika dibentuk secara garis besar.
Pendistribusian Tugas	Masing-masing tugas menjadi suatu prasyarat bagi tugas yang secara langsung berada di atasnya. Tugas yang berlangsung secara bersamaan dengan tugas lain, harus berada pada tingkatan yang sama di dalam hirarki.	Jika menggunakan format flowchart untuk melakukan analisis procedural, sebagian tugas di dalam flowchart dapat dipecah kedalam suatu format garis besar jika tugas tersebut memiliki subtask-subtask

Syarat	Perlunya mendaftar semua asumsi yang menjadi prasyarat pada dasar hirarki	Harus memiliki start dan akhir, semua tugas harus terhubung oleh tanda panah, dan lambang keputusan yang mengarah keluar hanya dapat berarti yes/no saja.
--------	---	---

2.1.2 Konsep Umum Hirarki (Tingkatan)

Sebuah hirarki merupakan organisasi antar unsur–unsur, dengan hubungan prasyarat, juga menunjukkan hubungan hirarkis diantara tugas–tugas, sebagai contoh seperti gambar 2.4 di bawah ini, *Task 4* hanya bisa di kerjakan bila *tasks* di bawahnya selesai, *task 3* baru bisa dikerjakan setelah *task 1* dan *task 2* selesai dikerjakan [5].



Gambar 2.4 Konsep Hirarki

Ada beberapa langkah utama untuk membangun sebuah hirarki [1]:

1. Cluster (pengelompokan) tugas-tugas yang ada.
 - Untuk memasukkan ke dalam suatu kelompok, parameter yang digunakan adalah tugas–tugas yang memiliki persamaan tujuan.

- Tiap tugas harus termasuk dalam paling tidak salah satu kelompok, tetapi sebuah tugas bisa jadi juga berada dalam beberapa kelompok.
 - Memberikan nama kepada kelompok tersebut sesuai dengan hubungan yang muncul dari tugas atau fungsi yang dianalisis.
 - Inisial kelompok dapat bersifat sementara, karena komposisi dari grup tersebut bisa berubah sebagai akibat dari keputusan yang diambil kemudian.
 - Jangan ragu untuk mengelompokkan ulang tugas-tugas ketika dipandang perlu.
2. Atur tugas-tugas dalam tiap grup untuk menunjukkan relasi hirarki untuk mempelajarinya.
- Saat relasi prasyarat muncul, evaluasi ulang relasi antar tugas apakah tugas superordinat ditampilkan jika tugas subordinat tidak dapat ditampilkan.
 - Level yang lebih rendah harus terelasi secara integral terhadap level yang lebih tinggi.
 - Tipe-tipe umum dari tujuan tugas harus cocok secara horizontal.
 - Bandingkan dengan pokok permasalahan lain yang lebih bagus untuk menentukan tingkat ketelitian hirarki.
 - Langkah ini dilakukan secara bersamaan dengan langkah 1 dan 2

2.1.3 Konsep Umum Analisis Tugas

Konsep tugas bisa mempunyai banyak arti, baik dari keseluruhan definisi pekerjaan hingga kepada operasi tunggalnya.

Terdapat tiga level konsep struktur :

- **Tujuan** (*Goals*)
- **Tugas** (*Tasks*)
- **Tindakan** (*Actions*)

Suatu Tujuan dapat digambarkan sebagai sesuatu yang ingin dicapai oleh seseorang. Untuk mencapai suatu tujuan, user umumnya memerlukan suatu rencana yang melibatkan satu set tugas untuk dilakukan dalam rangka menjangkau tujuan tersebut [5].

Suatu Tugas menjadi rangkaian tindakan atau aktivitas yang diperlukan untuk mencapai suatu tujuan. Suatu tugas juga merupakan satu set struktur aktivitas. Tugas dapat melibatkan pemecah masalah (*problem solving*) atau memilih antara tindakan alternatif (atau sub tugas) [5].

Suatu Tindakan merupakan suatu bentuk realisasi dari tugas sederhana yang tidak memerlukan suatu teknik pemecahan masalah. Suatu tindakan tidak mempunyai struktur, hanya bersifat melakukan sesuatu dari penugasan yang diberikan[5].

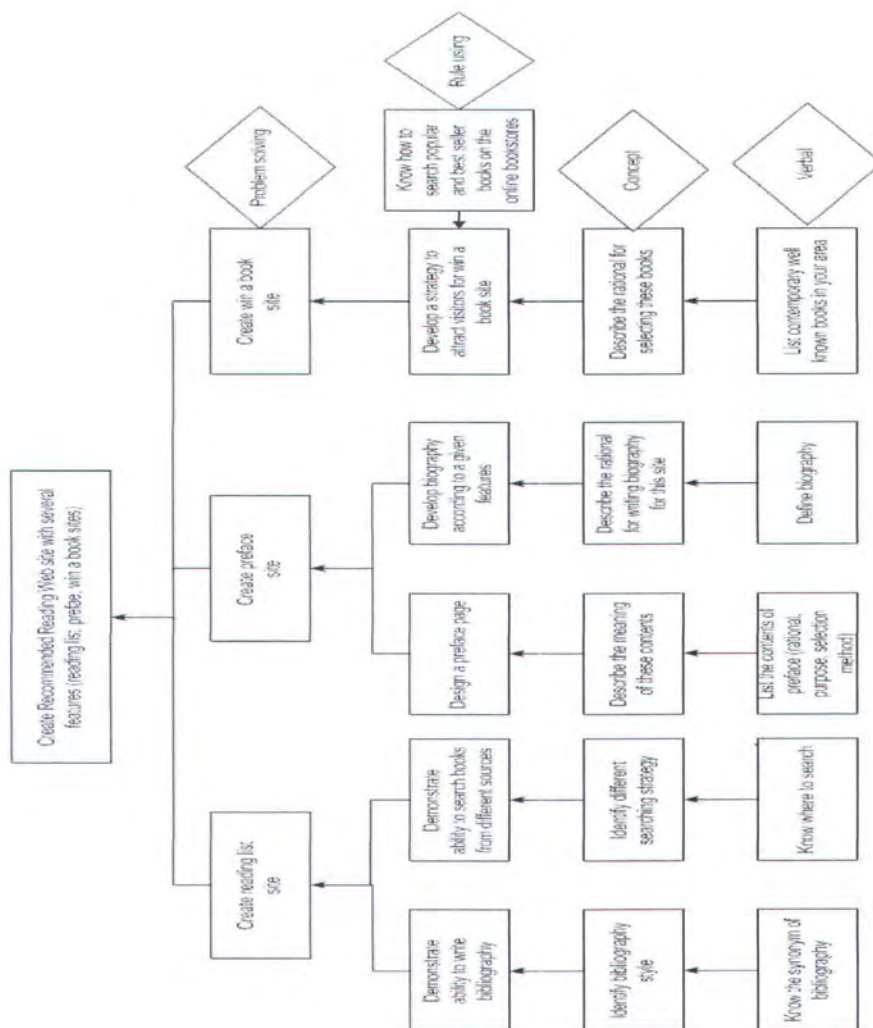
Ada beberapa langkah umum dalam menganalisis tugas [5] , yaitu :

1. Tujuan → Apa yang harus dilakukan
2. Memilih Alat atau Metode → Menentukan tugas di dalam bentuk syarat-syarat yang kongkrit

3. Tugas → Aktivitas untuk mencapai tujuan dengan menggunakan alat atau metode tersebut
4. Subtugas → Komponen suatu tugas
5. Tindakan → Tugas sederhana

2.1.4 Contoh Penggunaan Hierarchical Task Analysis (HTA) pada Pembuatan Reading Web Site

Gambar 2.5 di bawah ini merupakan contoh penggunaan HTA pada pembuatan *reading web site* oleh Khomar Khan pada tahun 1999



**Gambar 2.5 Contoh HTA : oleh Komar Khan
from EDCI 705, Fall 1999**

Dalam menyelesaikan kasus pembuatan *reading web site* di atas Khomar Khan menggunakan metode HTA, Adapun proses pembuatan *reading web site* dengan menggunakan metode HTA adalah sebagai berikut :

1. Memulai Analisis (*Starting the Analysis*)

Ada tiga langkah dalam memulai analisis, yaitu :

- a. Menentukan tugas utama, dalam kasus ini tugas utamanya adalah pembuatan *reading web site*.
- b. Membagi menjadi beberapa sub tugas. Khomar Khan membagi menjadi tiga sub tugas utama yaitu : *create reading list site*, *create preface site* dan *create win a book site*. Masing-masing sub tugas tersebut mempunyai sub-sub tugas di bawahnya , untuk lebih jelasnya dapat dilihat pada gambar 2.5
- c. Menggambarkan sub-sub tugas tersebut sebagai layer/lapisan-lapisan. Ada empat lapisan yang digunakan dalam pembuatan *reading web site ini*, yaitu :
 1. *Problem Solving* : Pengelompokan berdasarkan penyelesaian masalah yang ada.
 2. *Rule Using* : Pengelompokan berdasarkan aturan yang digunakan.
 3. *Concept* : Pengelompokan berdasarkan konsep yang dibuat dalam menyelesaikan tugas.
 4. *Verbal* : Pengelompokan berdasarkan penyelesaian tugas secara teknis.

2. Kemajuan Analisis (*Progressing Analysis*)

Ada dua langkah dalam proses ini, yaitu :

- a. Menentukan detail-detail pada tiap tingkatan. Dari tiga sub tugas utama yang ada dibagi lagi menjadi beberapa sub tugas hingga tugas tersebut tidak dapat lagi dibagi menjadi sub tugas. Gambar dari pembagian ini dapat dilihat pada gambar 2.5
- b. Menentukan pada tiap-tiap tugas apakah analisisnya dapat dilanjutkan atau tidak. Dalam kasus ini layer paling bawah adalah layer *verbal*, maka tugas yang termasuk dalam kategori layer verbal sudah tidak dapat lagi dibagi menjadi sub tugas, sebagai contoh tugas *know the synonym of bibliography* yang termasuk dalam layer verbal sudah tidak dapat lagi dibagi menjadi sub tugas.

3. Penyelesaian Analisis (*Finalizing the Analysis*)

Ada dua langkah dalam menyelesaikan analisis, yaitu :

- a. Memeriksa hasil dekomposisi apakah sudah mencakup semua alternatif. Dalam hal ini dilakukan pengecekan apakah hasil dekomposisi sudah seluruhnya tercakup sesuai dengan sub tugas utamanya.
- b. Melakukan evaluasi terhadap hasil dekomposisi. Adapun evaluasi dari dekomposisi sebagai berikut :
 - Untuk menyelesaikan tugas *reading list* dibagi dalam beberapa bagian, yaitu :
 1. Tingkat *verbal* : mengetahui sinonim dari struktur bahasa kamus/bibliograf dan mengetahui dimana tempat untuk *searching*.

2. Tingkat konsep : mengidentifikasi struktur bahasa kamus dan mengidentifikasi perbedaan strategi *searching*.
 3. Tingkat penggunaan aturan : mendemonstrasikan kemampuan untuk menulis bibliograf dan mendemonstrasikan kemampuan untuk melakukan pencarian dari sumber yang berbeda.
 4. Tingkat penyelesaian masalah : pembuatan *reading list site*
- Untuk menyelesaikan tugas *create reading list site* dibagi dalam beberapa bagian:
1. Tingkat *verbal* : mendefinisikan biografi
 2. Tingkat konsep : menjelaskan pengertian dari beberapa *content* dan menjelaskan secara rasional kegunaan penulisan biografi untuk *web site* ini.
 3. Tingkat penggunaan aturan : mendesain sebuah *preface page* dan membangun biografi berdasarkan *feature* yang telah diberikan.
 4. Tingkat penyelesaian masalah : pembuatan *preface site*.
- Untuk menyelesaikan tugas pembuatan *win book site* dibagi dalam beberapa bagian yaitu :
1. Tingkat *verbal* : Mendata buku-buku yang terkenal di daerah-daerah.
 2. Tingkat konsep : Menjelaskan secara rasional mengenai pemilihan buku-buku tersebut.

3. Tingkat penggunaan aturan : Membuat strategi untuk menarik pengunjung mengakses *win book site*.
4. Tingkat penyelesaian masalah : Pembuatan *win book site*.

2.2 Hubungan Hierarchical Task Analysis (HTA) dengan Report Designer

HTA merupakan sekumpulan tugas dan subtugas yang tersusun secara hirarki dan perencanaan/strategi yang melukiskan semua kondisi dari pekerjaan tugasnya. Pelevelan tugas menunjukkan hirarki tugas yang tersusun secara berjenjang. Dalam pembuatan aplikasi sistem informasi dengan menggunakan HTA tentunya tidak terlepas dengan pembuatan *report*. Hal ini dikarenakan *report* merupakan ringkasan dari data yang diolah oleh sistem informasi dan dilaporkan kepada pengguna dan dibuat berdasarkan kebutuhan pengguna. *Report* yang dibuat dalam sistem informasi ini dimaksudkan untuk memberikan informasi kepada pihak yang membutuhkan atau pihak manajemen yang digunakan dalam menentukan kebijakan-kebijakan yang akan diambil. Dalam pembuatan *report* tersebut tentunya diperlukan aplikasi *report designer* untuk mendesain *report* yang baik.

Report Designer pada umumnya digunakan untuk membuat suatu desain *report* yang baik. *Report designer* biasanya digunakan oleh pihak pengembang software dalam membuat suatu desain *report* yang baik dan tepat. *Report designer* yang ada sekarang pada umumnya berbentuk aplikasi yang mengkhususkan pada pembuatan *report*. *Fields* yang ada dalam *report designer* didapat dari hasil *query*

baik itu dari hasil *select*, *view* atau *store procedure*. *Fields* tersebut lalu digunakan untuk menentukan data-data yang perlu diambil untuk pembuatan suatu *report*. Adapun contoh *report designer* yang ada sekarang ini adalah *Crystal Report 10*.

Dalam proses HTA ini *report designer* digunakan untuk menunjang kinerja dari aplikasi pembuatan dan perancangan HTA itu sendiri, dalam hal ini *report designer* hanya digunakan untuk merancang desain laporan dari sebuah aplikasi sistem informasi dimana nanti hasil dari *report designer* disimpan dalam sebuah basis data HTA, yang kemudian basis data tersebut akan dibaca oleh sebuah aplikasi sistem informasi yang di buat dengan menggunakan metode HTA.

Report designer ini hanya bisa digunakan atau dibuat apabila sudah ada basis data HTA, karena dalam proses *report designer* untuk pengambilan *fields* dari sebuah hasil *query* diambil dari basis data HTA. Tetapi dalam *report designer* ini dapat ditambahkan fungsi yang digunakan dalam pembuatan aplikasi sistem informasi. Fungsi tersebut dideklarasikan dalam program yang membaca basis data HTA, karena bila fungsi tersebut digunakan dalam *report designer* tetapi fungsi tersebut tidak dideklarasikan terlebih dahulu maka akan menyebabkan *error* pada hasil *report*.

2.2 SQL Server

Dalam sistem basis data Client/Server, sebuah basis data berada pada sebuah/beberapa komputer pusat yang disebut dengan server, yang biasa dipakai bersama oleh banyak user yang disebut client. User mengakses server tersebut melalui sebuah aplikasi client yang dijalankan pada computer lain. User yang

berbeda-beda dan mengakses di banyak tempat mengakses data/informasi yang sama, yang akurat dan selalu update.

Di dalam sistem client server yang luas, ribuan user bisa saja terkoneksi pada sebuah instalasi SQL Server pada waktu yang sama. SQL Server memiliki perlindungan yang tinggi untuk menghadapi keadaan tersebut, seperti mencegah problem banyak user ingin mengupdate data yang sama pada waktu yang sama. SQL Server juga mengalokasikan resource yang tersedia pada waktu yang sama. SQL Server juga mengalokasikan resource yang tersedia dengan efektif, seperti memory, bandwidth jaringan, dan disk I/O di antara user yang berbeda-beda.

SQL Server dapat mengkonfigurasi sendiri secara dinamis untuk berjalan secara efisien berdasarkan resource yang tersedia pada client.

SQL Server menyediakan kepada basis data administrator beberapa sarana untuk mengelola sistem mereka, yakni :

1. SQL Server Enterprise Manager, yang merupakan bagian dari *Microsoft Management Console* (MMC). MMC mampu melakukan pengelolaan dari berbagai jenis server pada sebuah tempat, seperti MS Windows NT Services, Microsoft Internet Information Services, dan Microsoft SNA Servers. SQL Server Enterprise Manager menyajikan semua obyek SQL Server di dalam sebuah bentuk hirarki pohon dengan tampilan grafis yang mudah di gunakan.
2. SQL Server Agents, digunakan untuk mendefinisikan atau menjadwalkan suatu pekerjaan dengan basis data tertentu. Juga menyediakan sarana untuk memberitahu administrator ketika terjadi sesuatu, dan dapat juga diprogram untuk menjalankan aksi yang tepat.

3. SQL Server Profile, sebuah sarana bagi administrator untuk memonitor dan menganalisa lalu lintas jaringan dari dan menuju SQL Server.
4. SQL Server Performance Monitor, yang tergabung dengan performance Monitor milik Windows NT, merupakan alat bagi administrator untuk memonitor dan menggambarkan dalam bentuk grafik pendayagunaan SQL Server.
5. Index Tuning Wizards, menganalisis bagaimana sebuah atau lebih pernyataan SQL menggunakan index yang ada pada sekumpulan tabel.

SQL Server mendukung terhadap basis data API secara umum, seperti ADO, OLEDB, ODBC dan Embedded SQL for C, sehingga bisa terhubung dengan berbagai aplikasi Client yang berbeda-beda, baik yang dibuat dengan Visual Basic, Delphi, Visual C++, ASP dan sebagainya.

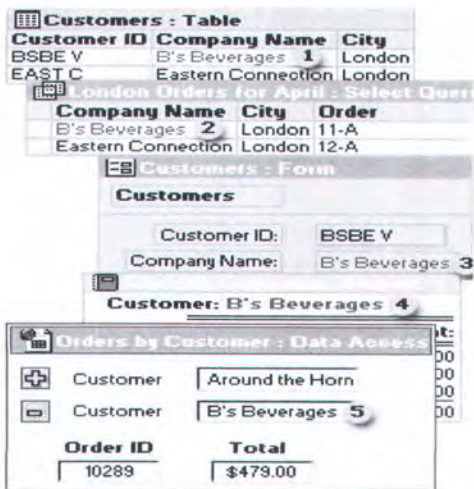
2.3 Microsoft Access

Microsoft Access merupakan suatu aplikasi yang dapat membantu untuk membuat aplikasi basis data dalam waktu yang relative singkat. Microsoft Access biasanya digunakan untuk pembuatan aplikasi-aplikasi yang kecil, misalnya program untuk penjualan, personalia, dsb.

Dengan menggunakan Microsoft Access, manajemen informasi dari suatu file basis data dapat dilakukan dengan menggunakan obyek-obyek dalam basis data Access:

- Table : menyimpan data
- Query : menemukan data yang dicari
- Form : melihat, menambah, dan mengubah data pada table

- Report : menganalisa data didalam layout khusus



Keterangan gambar:

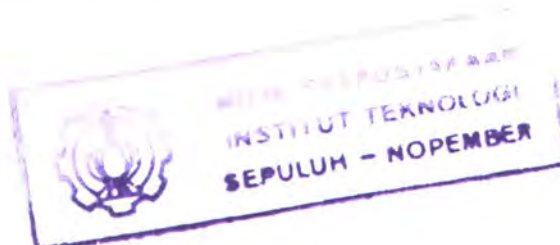
1. menyimpan data pada table
2. menampilkan data dalam suatu *query*
3. menampilkan data dalam suatu report/laporan
4. menampilkan data dalam data access page

Gambar 2.6 Obyek Basis Data Access

2.4 Visual Basic

Visual Basic merupakan salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an.

Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP).



Dalam Visual Basic, dikenal beberapa istilah dasar diantaranya: *Object* (komponen di dalam sebuah program), *Property* (karakteristik yang dimiliki oleh obyek), *Method* (aktivitas yang dapat dilakukan oleh obyek), *Event* (kejadian yang dapat dialami oleh obyek).

BAB III

PERANCANGAN SISTEM

Bab ini akan menjelaskan mengenai proses desain dari sistem perangkat lunak yang akan dibuat. Proses desain sistem dalam bab ini akan dibagi menjadi 5 tahapan: deskripsi umum sistem, spesifikasi kebutuhan sistem, perancangan proses, perancangan data, dan perancangan antarmuka.

3.1 Deskripsi Umum Sistem

Aplikasi yang akan dibuat dalam tugas akhir ini merupakan *report designer* yang digunakan sebagai alat bantu perancangan *hierarchical task analysis*.

Skenario proses bisnis yang terjadi dalam sistem report desainer ini dapat dijelaskan secara singkat sebagai berikut:

- Pihak pengembang aplikasi menentukan proses-proses bisnis yang ada di dalam aplikasi sistem informasi yang dibangun.
- Pihak pengembang aplikasi membuat basis data dengan menggunakan aturan *hierarchical task analysis*, dalam hal ini proses bisnis yang ada di simpan dalam basis data terpisah dengan basis data aplikasi demi keamanan.
- Pihak pengembang aplikasi membuat aplikasi tersendiri yang khusus untuk menangani report dari aplikasi sistem informasi yang dibangun dengan berdasarkan pada basis data *hierarchical task analysis*.

Pada Tugas Akhir kali ini akan dibuat sebuah *report designer* sebagai alat bantu perancangan hierarchical task analysis yang mampu merepresentasikan proses bisnis tersebut. Dalam perancangan report desainer ini harus sesuai dengan kode akses yang ada dalam basis data *hierarchical task analysis*.

Dalam pembuatan aplikasi simulasi ini, ada beberapa elemen yang akan berperan, antara lain:

- **Antarmuka (*interface*).**

Elemen ini merupakan tampilan yang berinteraksi langsung dengan pengguna (*end user*), berguna untuk menampung inputan dan menampilkan hasil pengolahan report. Antarmuka dalam aplikasi ini berupa desain untuk merancang bentuk laporan yang digunakan dalam aplikasi sistem informasi yang dibangun.

- **Basis Data (*Basis data*).**

Elemen ini merupakan simpanan kumpulan data yang dibutuhkan sistem. Basis data pada aplikasi ini meliputi kumpulan data yang terkait dengan *query* yang dibutuhkan dalam pembuatan report aplikasi, proses bisnis yang ada dalam aplikasi sistem informasi, desain aplikasi, dan desain *report* dari aplikasi sistem informasi yang dibangun.

- **Program aplikasi (*Application Programming*).**

Elemen ini merupakan program yang mengolah input dari antarmuka sehingga menghasilkan output yang diinginkan. Untuk input berupa desain dari *report* yang diinginkan berdasarkan kode akses yang berisi *query* untuk kebutuhan

dalam mendesain *report*. Sedangkan hasil dari desain tersebut dapat di cetak melalui sistem aplikasi sistem informasi yang dibangun.

3.2 Spesifikasi Kebutuhan Sistem

Spesifikasi kebutuhan sistem menjelaskan mengenai pengguna sistem dan hak aksesnya, masukan dan keluaran sistem, dan komunikasi datanya.

3.2.1 Pengguna Aplikasi

Pengguna aplikasi ini adalah pihak pengembang aplikasi sistem informasi yang membangun aplikasi sistem informasi dengan menggunakan aturan-aturan yang ada dalam *hierarchical task analysis* (HTA).

Pihak pengembang dapat menggunakan aplikasi ini untuk membantu dalam pembuatan desain *report* yang diinginkan dalam aplikasi sistem informasi. Dalam pembuatan desain *report* tersebut, pihak pengembang harus mengacu pada kode akses yang telah dibuat dalam basis data HTA.

3.2.2 Masukan dan Keluaran Sistem

Masukan sistem terdiri atas sejumlah data yang diperlukan untuk memenuhi kebutuhan sistem yang ada. Sedang keluaran sistem terdiri atas sejumlah data yang dihasilkan oleh proses yang terjadi di dalam sistem.

Dalam sistem aplikasi kali ini, data masukan sistem meliputi kode akses yang digunakan untuk report yang di buat dan informasi data dari komponen yang digunakan dalam desain *report* tersebut beserta atribut-atributnya.

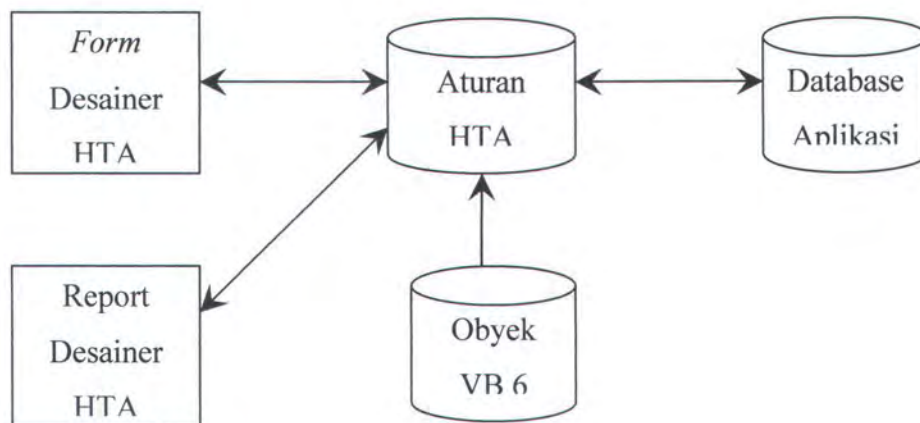
Masukan kode akses dan informasi komponen merupakan permintaan layanan ke sistem. Proses pengolahan akan dilakukan sesuai dengan data masukan sistem. Keluaran sistem akan menghasilkan desain *report* yang digunakan dalam

pembuatan aplikasi sistem informasi yang dibuat. *Design report* tersebut juga dapat disimpan dan dimodifikasi sesuai kebutuhan report konsumen.

3.2.3 Hubungan Hirarchical Task Analysis (HTA), Report Designer dan Aplikasi Sistem

Hubungan sistem aplikasi *report designer* ini dengan aplikasi sistem informasi yang dibangun berada pada tingkatan proses bisnis yang di gunakan dalam pembuatan aplikasi sistem informasi. Aplikasi *report designer* mempunyai peranan penting dalam pembuatan desain *report* aplikasi sistem informasi yang dibuat. Keterkaitan *report designer* dengan HTA serta aplikasi sistem dapat dilihat pada gambar 3.1

Sistem *report designer* ini akan banyak digunakan oleh pihak pengembang aplikasi untuk membantu dalam pembuatan *report* yang ada dalam aplikasi sistem informasi yang dibangun. Aplikasi ini juga akan mempermudah dalam pembuatan *report* yang dibutuhkan oleh pemakai aplikasi sistem informasi.



Gambar 3.1 Arsitektur Sistem

Dalam sistem aplikasi ini basis data HTA yang di representasikan sebagai aturan HTA berperan sebagai penghubung antara basis data khusus untuk menyimpan data-data yang ada dalam sistem informasi yang dibangun dengan basis data yang menyimpan proses bisnis yang terjadi dalam sistem aplikasi sistem informasi.

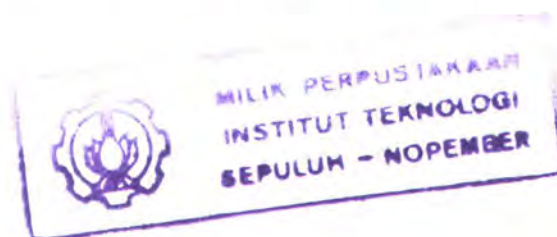
Pembuatan basis data HTA dilakukan terpisah dengan basis data aplikasi dikarenakan dalam basis data HTA terdapat semua proses bisnis dalam perancangan aplikasi itu sendiri, selain itu pemisahan ini dilakukan demi keamanan aplikasi yang dibuat. Bila terjadi kerusakan dalam basis data aplikasi sistem informasi maka proses bisnis yang ada dalam aplikasi tersebut khususnya desain *report* tidak ikut mengalami kerusakan juga.

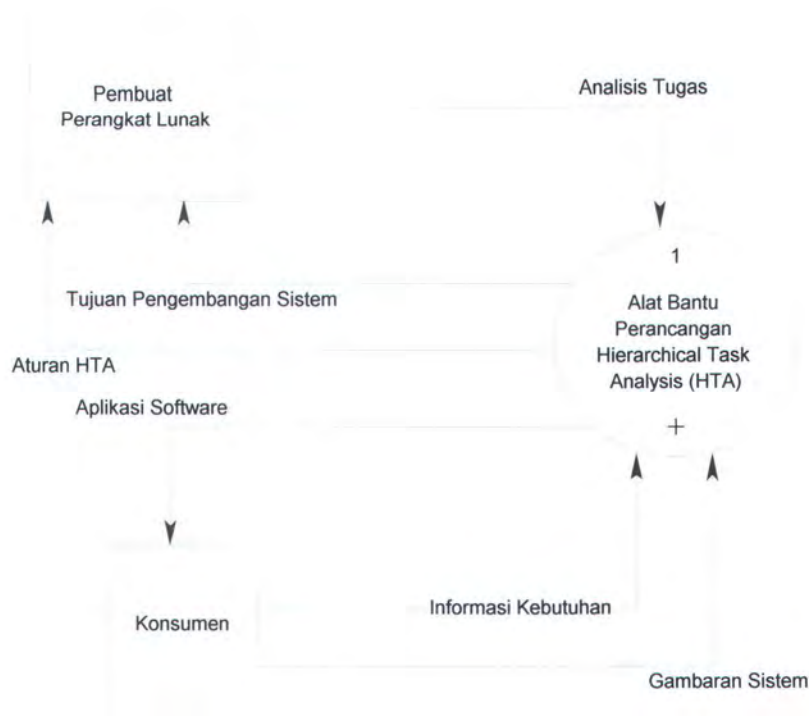
3.3 Perancangan Proses

Perancangan Proses digunakan untuk menggambarkan sejumlah proses terstruktur dalam sistem, berorientasikan pada aliran proses yang terjadi. Dalam perancangan proses kali ini digunakan *Data Flow Diagram* (DFD) yang dibuat dengan Power Designer 6.

3.3.1 DFD Level 0

DFD level 0 menjelaskan gambaran umum mengenai sistem yang akan dibuat, gambar dari DFD level 0 dapat dilihat pada gambar 3.2 . Ada 2 entitas yang terlibat dalam sistem ini yaitu: Konsumen dan Pembuat Perangkat Lunak. Pada tabel 3.1 dijelaskan mengenai definisi masing-masing entitas tersebut.





Gambar 3.2 DFD Level 0

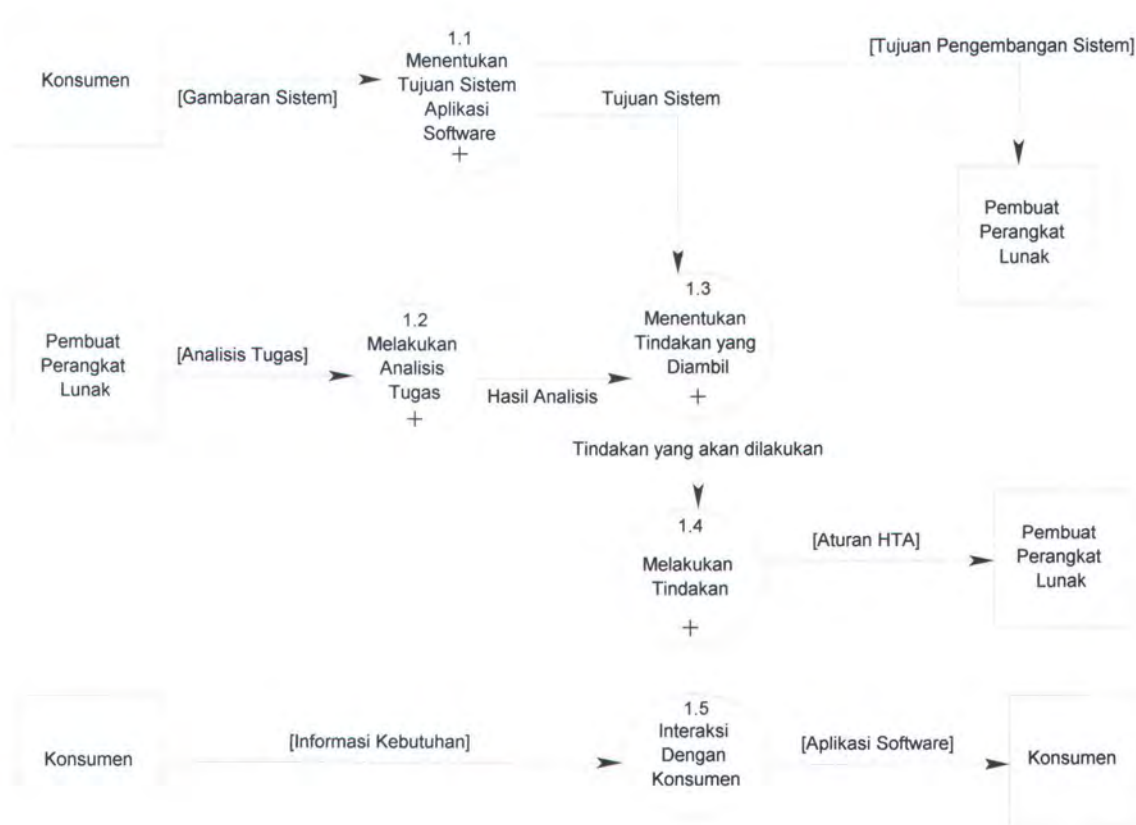
Tabel 3.1 Definisi Entitas DFD Level 0

No	Entitas	Definisi
1	Konsumen	Pemesan dan pengguna dari aplikasi sistem informasi yang dibangun dengan menggunakan aturan HTA
2	Pembuat Perangkat Lunak	Pembuat aplikasi sistem informasi dengan menggunakan aturan HTA

Pihak konsumen memberikan informasi mengenai gambaran sistem dan kebutuhan-kebutuhan apa saja yang diperlukan dalam sistem informasi yang akan dibangun. Pihak konsumen akan mendapatkan aplikasi sistem informasi sesuai

dengan informasi kebutuhan yang diberikan. Pihak pembuat perangkat lunak melakukan analisis tugas dan tindakan yang diambil untuk mencapai tujuan sistem

3.3.2 DFD Level 1



Gambar 3.3 DFD Level 1

Pada DFD level 1 yang digambarkan pada gambar 3.3 diatas terdapat 5 proses utama, yaitu:

1. Proses 1.1 : Menentukan Tujuan Sistem Aplikasi Perangkat Lunak

Pada proses ini konsumen diminta memberikan gambaran lengkap mengenai tujuan sistem aplikasi sistem informasi yang akan dibuat. Proses ini membutuhkan masukan sistem berupa gambaran sistem. Pemrosesan

selanjutnya menghasilkan tujuan sistem yang akan digunakan untuk analisis sistem dan tindakan yang perlu dilakukan.

2. Proses 1.2 : Melakukan Analisis Tugas

Pada proses ini developer software akan menganalisis tugas-tugas yang akan didistribusikan. Hasil analisis tugas nantinya akan dianalisis lebih lanjut pada proses Penyatuan Tujuan Analisis dan Tindakan.

3. Proses 1.3 : Menentukan Tindakan yang Diambil

Pada proses ini hasil gambaran konsumen akan sistem yang akan dibuat, dan analisis pendistribusian tugas dari developer akan disatukan dan dianalisis lebih lanjut menghasilkan suatu tindakan

4. Proses 1.4 : Melakukan Tindakan

Proses ini menindaklanjuti tindakan, hasil dari analisis menjadi sebuah *design report* yang digunakan dalam aplikasi sistem informasi yang dibangun.

5. Proses 1.5 : Interaksi Dengan Konsumen

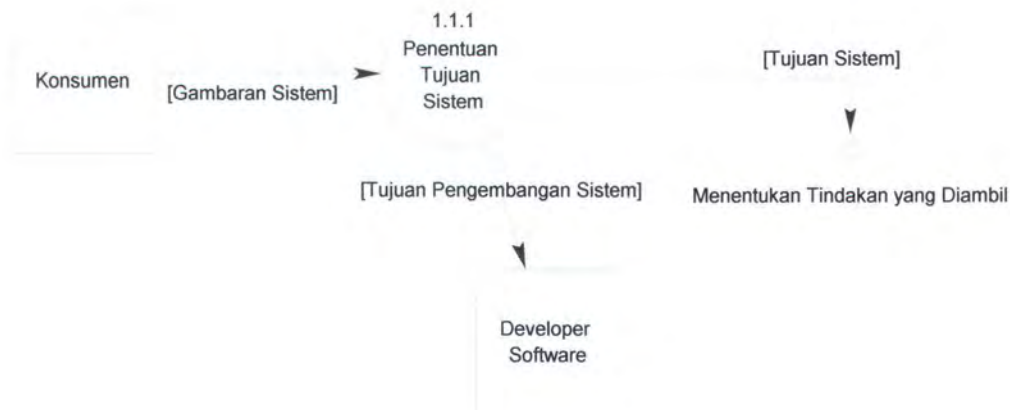
Proses ini digunakan untuk berinteraksi dengan konsumen / pengguna software. Proses ini membutuhkan masukan sistem berupa informasi kebutuhan sistem. Pemrosesan selanjutnya menghasilkan aplikasi sistem informasi.

3.3.3 DFD Level 2

Pada aplikasi ini, semua proses pada DFD level 1 diturunkan ke DFD level

2 berikut ini:

3.3.3.1 Menentukan Tujuan Sistem Aplikasi

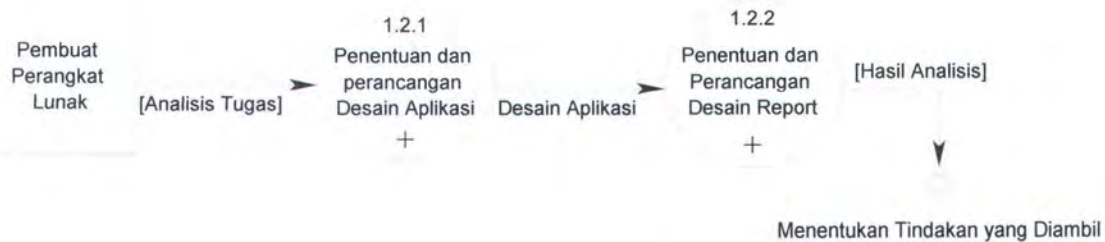


Gambar 3.4 DFD Level 2 – Menentukan Tujuan Sistem Aplikasi

Proses ini digunakan untuk menentukan tujuan sistem aplikasi yang akan dibuat. Masukan data berupa gambaran sistem didapatkan dari pihak konsumen. Dari gambaran sistem tersebut dapat ditarik suatu kesimpulan bentuk sistem aplikasi yang akan dibangun.

3.3.3.2 Melakukan Analisis Tugas

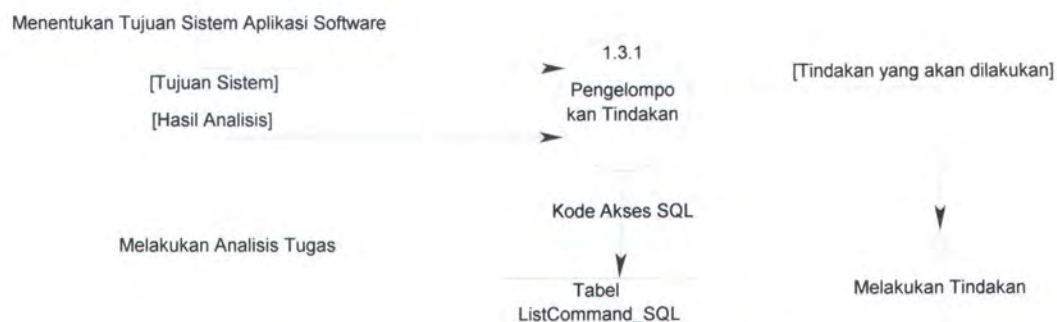
Proses ini digunakan untuk melakukan analisis tugas – tugas yang ada untuk pembuatan aplikasi sistem informasi. Gambar dari proses ini dapat dilihat pada gambar 3.5. Dalam proses penentuan dan perancangan desain aplikasi didapat masukan berupa analisis tugas yang kemudian menghasilkan desain aplikasi yang digunakan dalam proses perancangan dan *design report*.



Gambar 3.5 DFD Level 2 – Melakukan analisis tugas

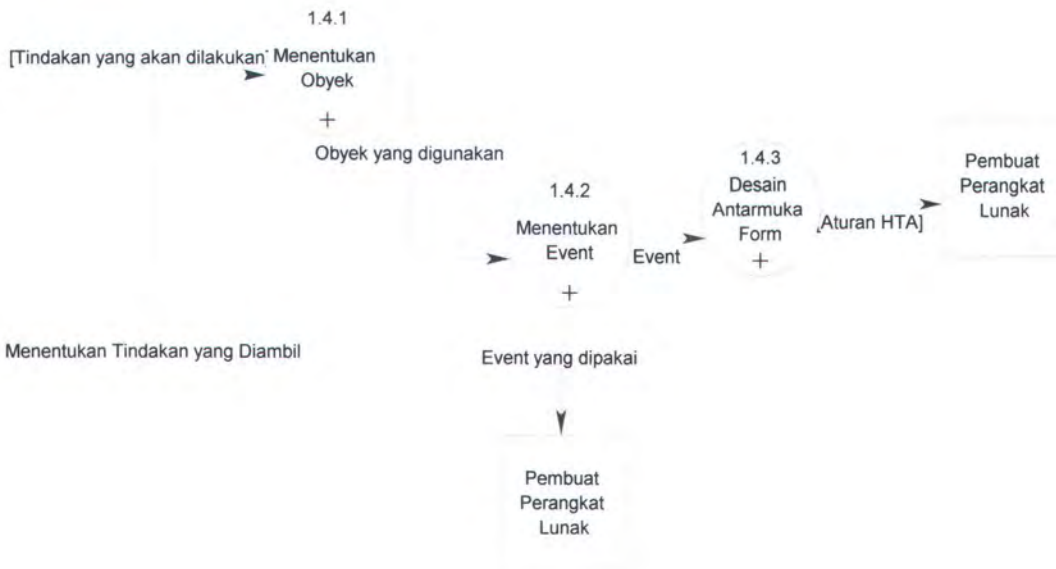
3.3.3.3 Menentukan Tindakan yang Diambil

Proses ini digunakan untuk mentukan tindakan yang diambil dengan cara mengelompokkan tindakan–tindakan yang memiliki kesamaan tujuan sehingga dapat mempercepat penyelesaian pembuatan aplikasi sistem informasi. Hasil dari sistem ini akan berupa kode akses–kode akses SQL yang berfungsi untuk mendapatkan data dari basis data aplikasi, kode akses tersebut juga digunakan sebagai kode akses dari form. Gambar dari proses ini dapat dilihat pada gambar 3.6 di bawah ini.



Gambar 3.6 DFD Level 2 –Menentukan Tindakan yang Diambil

3.3.3.4 Melakukan Tindakan



Gambar 3.7 DFD Level 2 –Melakukan Tindakan

Dari gambar 3.7 di atas dapat dilihat pada proses melakukan tindakan terdiri atas 3 proses, yaitu :

1. Proses Menentukan Obyek

Proses ini menentukan obyek – obyek dari visual basic yang akan digunakan untuk mendukung perancangan aplikasi sistem informasi. Proses ini memperoleh masukan tindakan yang dilakukan. Proses selanjutnya adalah menghasilkan obyek–obyek visual basic yang digunakan dalam aplikasi.

2. Proses Menentukan Event

Proses ini digunakan untuk menentukan event–event yang akan digunakan dari masing–masing obyek vusial basic.

3. Proses Desain Interface Form

Proses ini digunakan untuk mendesain tampilan form yang akan digunakan dalam aplikasi sistem informasi yang dibangun. Proses ini akan menghasilkan aturan HTA yang digunakan dalam aplikasi sistem informasi yang dibuat.

3.3.4 DFD Level 3

Pada aplikasi ini, semua proses pada DFD level 2 diturunkan ke DFD level 3 berikut ini:

3.3.4.1 Penentuan dan Perancangan Desain Aplikasi

Gambar DFD dari proses ini dapat dilihat pada gambar 3.8. Proses ini terdiri atas proses berikut ini :

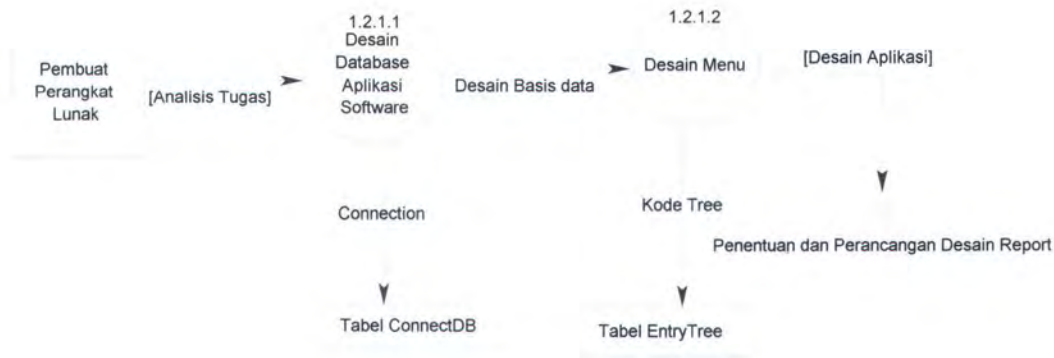
1. Proses Desain Basis data Aplikasi Software

Proses ini digunakan untuk mendesain basis data aplikasi. Masukan data proses ini berupa analisis tugas, untuk koneksi dengan basis data aplikasi digunakan Tabel ConnectDB yang menyimpan ConnectionString.

2. Proses Desain Menu

Proses ini digunakan untuk membuat menu-menu yang harus ada dalam aplikasi sistem informasi yang dibuat, hasil dari proses ini digunakan untuk proses selanjutnya berupa menu-menu yang dipakai, menu-menu yang digunakan disimpan dalam tabel EntryTree.





Gambar 3.8 DFD Level 3 – Penentuan dan Perancangan Desain Aplikasi

3.3.4.2 Penentuan dan Perancangan Desain Report

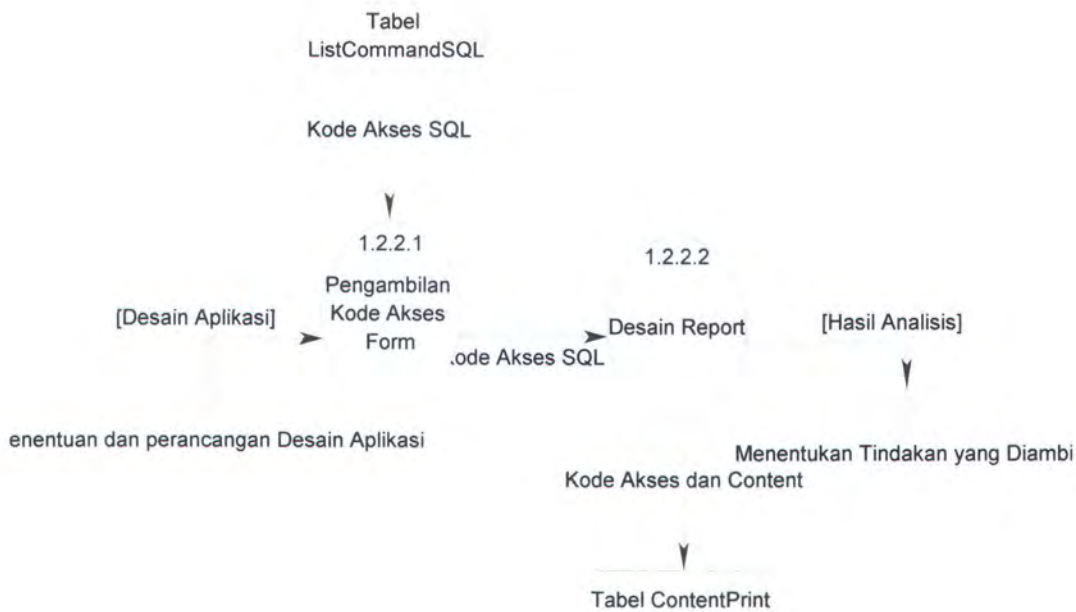
Gambar DFD dari proses ini dapat dilihat pada gambar 3.9. Proses ini terdiri atas proses berikut ini :

1. Proses Pengambilan Kode Akses Form

Proses ini digunakan untuk menentukan kode akses report yang akan dibuat dalam aplikasi sistem informasi yang dibangun. Proses ini memperoleh masukan data desain aplikasi dan Kode Akses SQL dari basis data (tabel List Command SQL). Proses ini menghasilkan kode akses yang digunakan untuk acuan proses desain *report*.

2. Proses Desain *Report*

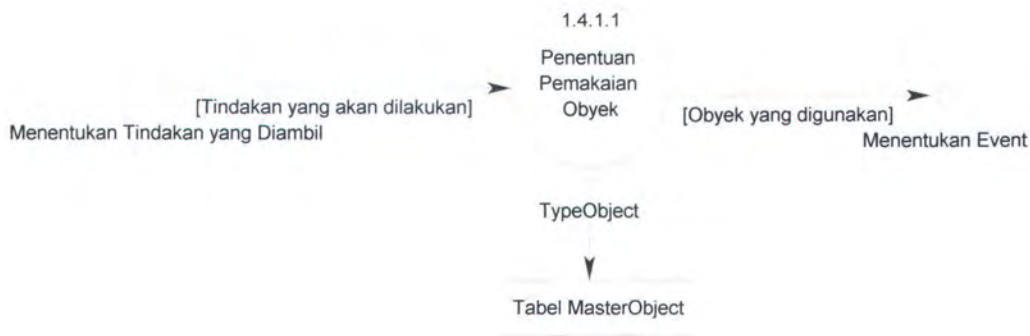
Proses ini merupakan proses untuk melakukan desain *report* dari data-data yang digunakan dalam aplikasi sistem informasi. Proses ini memperoleh masukan data kode akses dari proses sebelumnya. Pemrosesan selanjutnya akan menghasilkan desain *report* yang digunakan untuk proses berikutnya.



Gambar 3.9 DFD Level 3 – Penentuan dan Perancangan Desain report

3.3.4.3 Menentukan Obyek

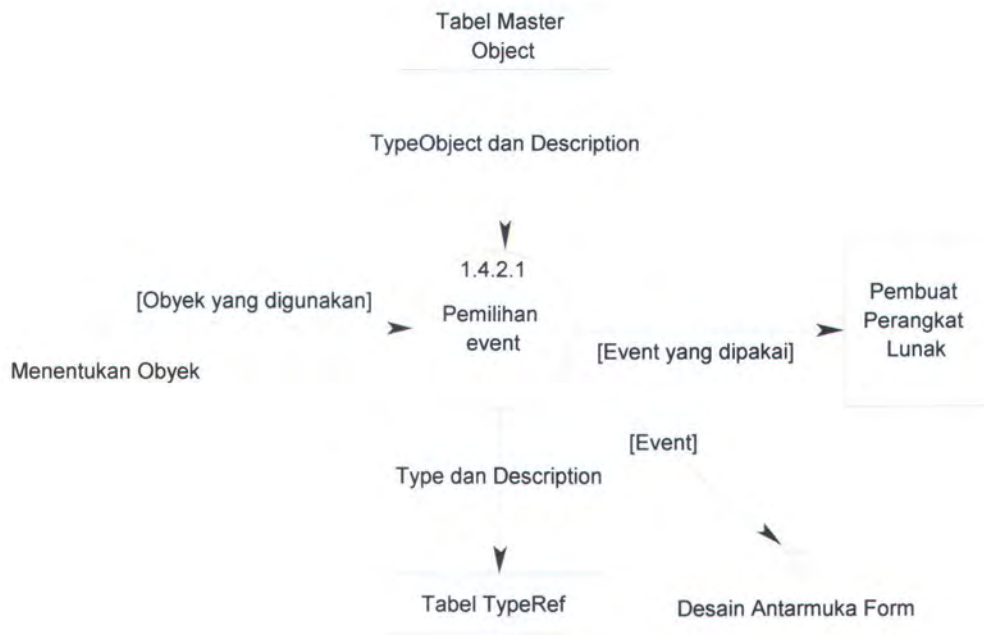
Proses ini menentukan obyek-obyek dari visual basic yang akan digunakan untuk mendukung perancangan aplikasi sistem informasi. Proses ini memperoleh masukan dari tindakan yang akan dilakukan. Proses selanjutnya adalah melakukan penyimpanan dalam basis data (tabel Master Object) dengan mengirimkan data Type Object. Selain itu hasil dari pemrosesan digunakan untuk proses selanjutnya yaitu untuk menentukan event yang digunakan dengan berdasar kepada obyek yang digunakan. Gambar dari Proses ini dapat dilihat pada gambar 3.10 di bawah ini.



Gambar 3.10 DFD Level 3 – Menentukan Obyek

3.3.4.4 Menentukan Event

Proses ini digunakan untuk menentukan event–event apa saja yang digunakan untuk mendukung pembuatan aplikasi sistem informasi. Proses ini memperoleh masukan data dari basis data (tabel Master Object) berupa Type Object dan dari hasil proses penentuan obyek visual basic yang digunakan. Hasil dari proses ini akan berupa type event dan deskripsinya yang disimpan dalam basis data (tabel Type Ref). Selain itu hasil dari proses ini juga digunakan untuk proses selanjutnya yaitu desain interface form. Gambar dari Proses ini dapat dilihat pada gambar 3.11 di bawah ini.



Gambar 3.11 DFD Level 3 – Menentukan Event

3.3.4.5 Desain Antarmuka Form

Gambar 3.12 di bawah ini merupakan gambaran dari proses desain antarmuka form. Proses ini terdiri atas proses berikut ini :

1. Proses Desain Form

Proses ini digunakan untuk menentukan desain form aplikasi sistem informasi. Proses ini memperoleh masukan berupa menu. Kemudian hasil-nya yang berupa setting form (tampilan form) disimpan dalam basis data (tabel refCmd) dan juga digunakan untuk proses integrasi antar Form.



**Gambar 3.12 DFD Level 3 – Desain Antarmuka
*Form***

2. Proses Integrasi Antarmuka Form

Proses ini digunakan untuk mengintegrasikan forms yang sudah dibuat menjadi satu kesatuan dalam satu project. Proses ini menghasilkan aturan HTA yang digunakan untuk merancang aplikasi sistem informasi.

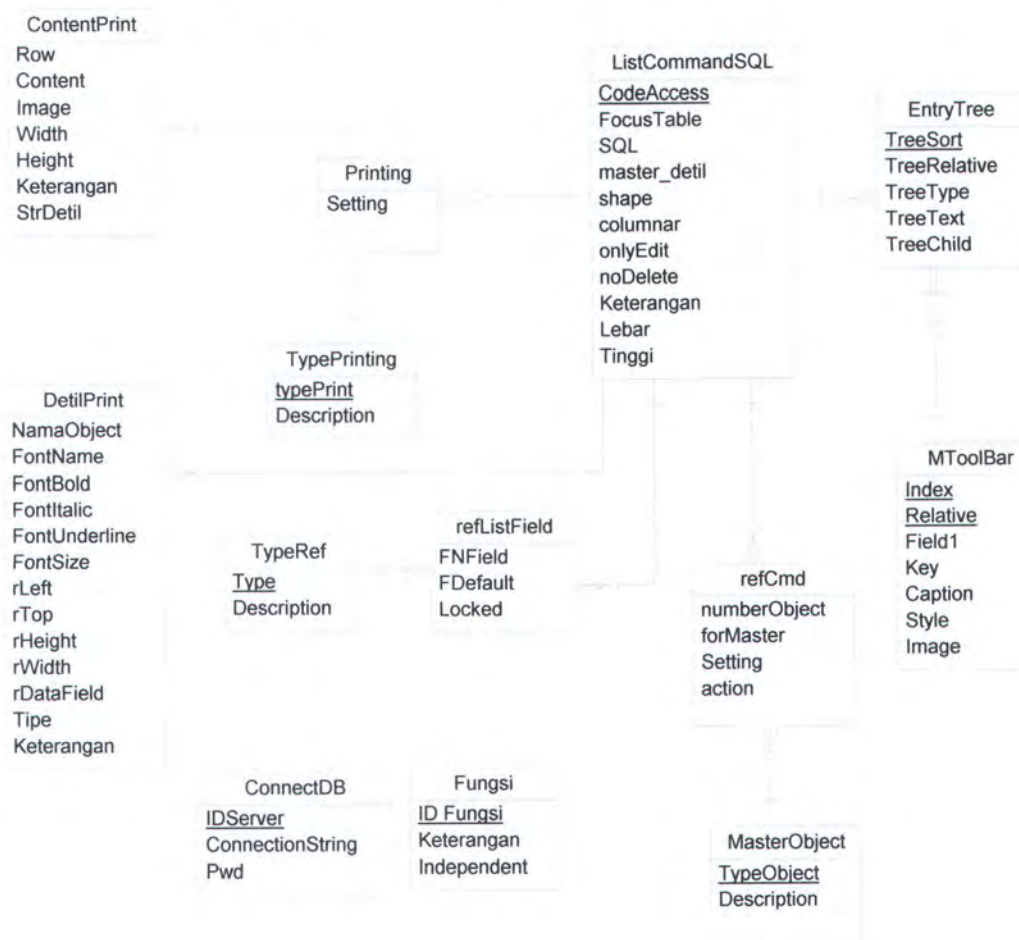
3.4 Perancangan Data

Perancangan data ini meliputi model data konseptual dan model data fisik. Model data konseptual merepresentasikan struktur logis basis data secara keseluruhan, yang berisi obyek data dengan atribut dan relasinya yang masih belum diimplementasikan dalam basis data fisik. Sedangkan model data fisik merepresentasikan struktur implementasi basis data, berisi obyek data yang akan diimplementasikan dalam basis data fisik.

3.4.1 Model Data Konseptual

Model data konseptual pada aplikasi ini merepresentasikan gambaran sistem, proses bisnis dan desain *report* yang di buat sesuai dengan aturan *hierarchical task analysis* (HTA).

Gambar 3.13 di bawah ini merupakan model data konseptual HTA yang digambar dengan menggunakan *tool* Power Designer 6:



Gambar 3.13 Diagram Data Konseptual – Sistem HTA

3.5 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan halaman aplikasi yang berinteraksi langsung dengan pengguna. Perancangan antarmuka pada aplikasi ini antarmuka pada desain *report*.

3.2.1 Antarmuka pada Desain Report

Antarmuka pada desain *report* ini terbagi menjadi enam bagian, yaitu :

1. Antarmuka komponen-komponen yang digunakan, pada antarmuka ini terdapat beberapa komponen yang dapat digunakan untuk pembuatan desain sistem informasi. Gambar dari rancangan antarmuka ini dapat dilihat pada gambar 3.14 di bawah ini.



Gambar 3.14 Antarmuka Komponen yang digunakan

Kode Akses	Tanda Digunakan	Keterangan Kode Akses

Tombol Digunakan

Tombol Untuk Edit Keterangan

Tombol Keluar

Gambar 3.16 Antarmuka Kode Akses *Form*

4. Antarmuka Fungsi, pada antarmuka ini pihak pengembang dapat menentukan fungsi–fungsi yang diperlukan dalam pembuatan suatu desain *report*. Gambar dari rancangan antarmuka ini dapat dilihat pada gambar 3.17

Nama Fungsi	Kegunaan	Penggunaan

Tombol untuk Digunakan

Tombol Keluar

Gambar 3.17 Antarmuka Fungsi yang Digunakan

5. Antarmuka Pengaturan Kertas dan *margin*, digunakan untuk menentukan ukuran *margin*. Gambar dari rancangan antarmuka ini dapat dilihat pada gambar 3.19 di bawah ini.

Margin

Top Bottom

Left Right

☐ Checkbox Pilihan untuk ditampilkan/tidak

Tombol Ok Tombol Batal

Gambar 3.18 Antarmuka Pengaturan Kertas dan Margin

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Pada bab IV ini akan dibahas mengenai implementasi dari rancangan sistem yang telah dibuat pada bab III. Bagian implementasi sistem kali ini meliputi: lingkungan implementasi, implementasi data, implementasi proses, dan implementasi antarmuka.

4.1 Lingkungan Implementasi

Pada bagian ini akan dijelaskan mengenai komponen-komponen yang diperlukan agar sistem dapat berjalan dengan baik. Komponen-komponen tersebut antara lain:

d. Microsoft Visual Basic 6.0

Pada implementasi ini, bahasa pemrograman yang dipakai untuk membuat alat bantu perancangan *Hierarchical Task Analysis (HTA) report designer* adalah Microsoft Visual Basic 6.

e. Microsoft Access 2000

Untuk menyimpan dan mengolah semua data HTA di aplikasi ini, penulis menggunakan Microsoft Access 2000.

f. Microsoft SQL Server 2000

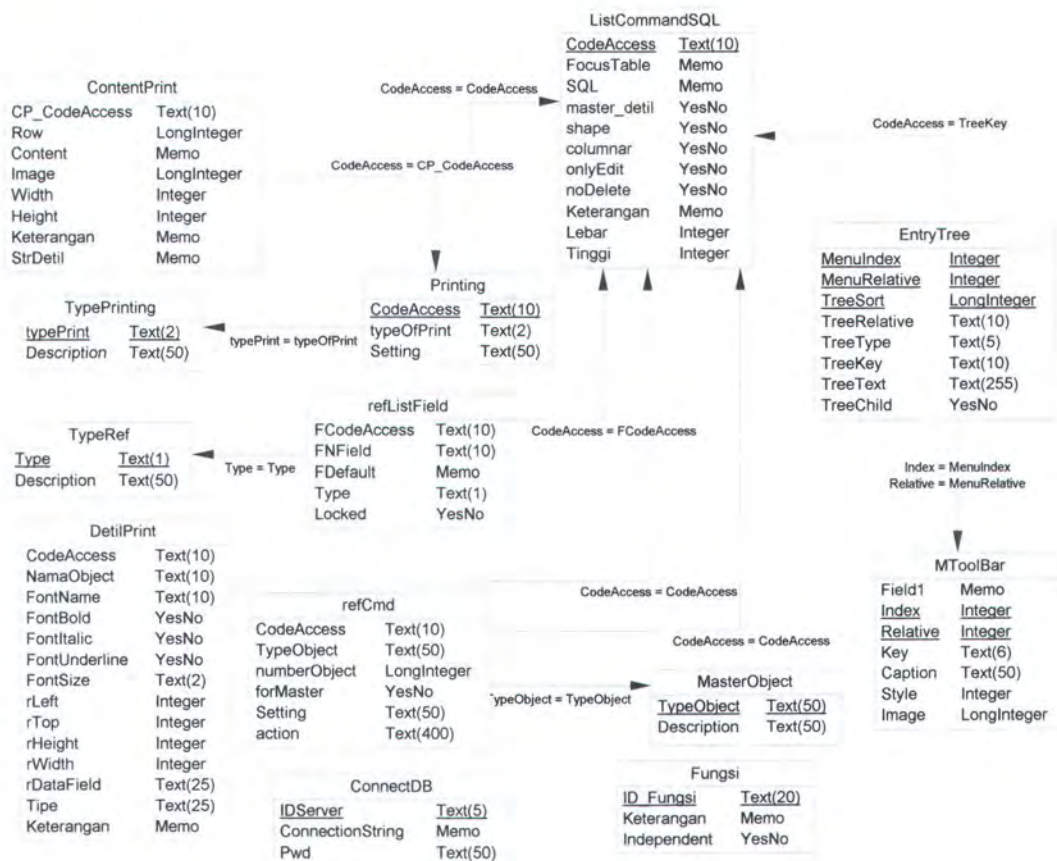
Untuk menyimpan dan mengolah data contoh aplikasi sistem informasi yang dibangun dengan menggunakan metode HTA.

4.2 Implementasi Data

Rancangan data konseptual yang telah dipetakan menjadi diagram diagram pada bagian perancangan sistem akan diimplementasikan ke dalam lingkungan basis data Microsoft Access 2000.

4.2.1 Model Data Fisik

Gambar 4.1 di bawah ini merupakan model diagram data konseptual yang dibuat dengan cara *me-generate* diagram data konseptual di atas (*tool* Power Designer 6 ini menyediakan fasilitas untuk melakukan *generate* model data konseptual).



Gambar 4.1 Diagram Data Fisik – Sistem HTA

Diagram data fisik ini menghasilkan tabel-tabel yang akan digunakan dalam implementasi aplikasi.

Pada sistem aplikasi HTA ini, tabel-tabel yang digunakan antara lain:

1. Tabel ConnectDB, digunakan untuk menyimpan server, stringkoneksi, dan password basis data aplikasi.
2. Tabel TypeRef, digunakan untuk menyimpan *events* visual basic yang digunakan dalam perancangan aplikasi sistem informasi.
3. Tabel TypePrinting, digunakan untuk menyimpan informasi lokasi pencetakan laporan.
4. Tabel RefListField, digunakan untuk menyimpan acuan *fields* yang ada dalam basis data sistem informasi pada komponen visual basic yang digunakan.
5. Tabel refCmd, digunakan untuk menyimpan data informasi pada komponen yang di gunakan dalam setiap kode akses.
6. Tabel Printing, di gunakan untuk menyimpan informasi letak file report khusus untuk pencetakan dalam microsoft word.
7. Tabel Mtoolbar, digunakan untuk meyimpan nama-nama menu yang ada dalam aplikasi sistem informasi.
8. Tabel masterObject, digunakan untuk menyimpan obyek-obyek visual basic yang digunakan dalam pembuatan aplikasi sistem informasi.
9. Tabel LoginScript, digunakan untuk menyimpan script sql pada saat login ke dalam aplikasi sistem informasi

10. Tabel Detil Print, digunakan untuk menyimpan kondisi obyek detil pada desain *report*
11. Tabel ListCommandSQL, digunakan untuk menyimpan *query* yang digunakan oleh masing – masing kode akses
12. Tabel EntryTree, digunakan untuk mengatur hierarki dari sebuah sub menu.
12. Tabel ContentPrint, di gunakan untuk menyimpan kondisi *design report* yang digunakan berdasarkan kode akses yang telah dibuat pada tabel ListCommandSQL.

Sedangkan untuk basis data alat bantu perancangan *report designer* terdapat sejumlah tabel yang digunakan antara lain :

- Tabel ListCommand SQL

Pada tabel 4.1 di bawah ini dapat dilihat tentang *fields* dan tipe data yang digunakan dalam tabel ListCommandSQL.

Tabel 4.1 Tabel ListCommandSQL

Nama Kolom	Tipe Data
CodeAccess	Varchar(10)
FocusTable	Text
SQL	Text
Master_Detil	Boolean
shape	Boolean
columnar	Boolean
onlyEdit	Boolean

noDelete	Boolean
Keterangan	Text
JenisKertas	Varchar(15)
Lebar	Integer
Tinggi	Integer

- Tabel ContentPrint

Pada tabel 4.2 di bawah ini dapat dilihat tentang *fields* dan tipe data yang digunakan dalam tabel ContentPrint.

Tabel 4.2 Tabel ContentPrint

Nama Kolom	Tipe Data
CP_CodeAccess	Varchar(10)
Row	LongInteger
Content	Text
Image	OLE Object
Width	Integer
Height	Integer
Keterangan	Text
strDetil	Text

- Fungsi

Pada tabel 4.3 di bawah ini dapat dilihat tentang *fields* dan tipe data yang digunakan dalam tabel Fungsi.

Tabel 4.3 Tabel Fungsi

Nama Kolom	Tipe Data
ID_Fungsi	Varchar(10)
Keterangan	Text
Independent	Boolean

4.3 Implementasi Proses

Bagian implementasi proses ini menjelaskan mengenai implementasi proses-proses sebagaimana yang rancangan DFD telah dibuat pada bab III. Dalam model perancangan tersebut, Pada sub bab implementasi proses ini penulis lebih mengkhususkan untuk menjelaskan perancangan DFD level 2 pada proses penentuan dan perancangan desain *report* (proses 1.2.2)

4.3.1 Penentuan dan Perancangan Desain *Report*

Proses ini digunakan untuk melakukan desain dan perancangan report yang dipakai dalam aplikasi sistem informasi. Pemrosesan ini menghasilkan *design report* yang digunakan dalam aplikasi sistem informasi yang dibangun.

Pengambilan Kode Akses Form

Proses ini digunakan untuk menentukan kode akses *report* yang akan dibuat dalam aplikasi sistem informasi yang dibangun. Kode akses tersebut tidak dapat

dirubah atau di hapus dari aplikasi ini dikarenakan kode akses tersebut merupakan kode yang digunakan pada aplikasi sistem informasi.

Sebelum membuat desain *report*, maka pihak pengguna aplikasi harus mengetahui terlebih dahulu kode akses dari suatu tampilan antarmuka dari aplikasi sistem informasi, sehingga tidak terjadi kekeliruan dalam pembuatan desain *report*. Untuk mengatasi ini kekeliruan dalam menentukan kode akses, penulis menyediakan *field* keterangan pada tabel ContentPrint yang berguna untuk menjelaskan kode akses dari antarmuka aplikasi sistem informasi. Pseudocode dari proses pengambilan kode akses form dapat dilihat pada gambar 4.2 di bawah ini

Private Sub AmbilKodeAkses()

Mengambil SQL dari tabel List Comand SQL \leftarrow *select * from listcommandsql where codeaccess='kode Akses'*

JmlKolom \leftarrow *GetKolom(rs!Sql)* *'Memisahkan Kolom'*

Mengecek Apakah dalam tabel ContentPrint sudah ada desain \leftarrow *select * from contentprint where cp_codeaccess='Kode Akses'*

if Recordset.Recordcount > 0 'Jika ada'

Tampilkan Desain

End if

Gambar 4.2 Pseudocode untuk Mengambil Kode Akses

Desain Report

Proses ini merupakan proses untuk melakukan *design report* dari data-data yang digunakan dalam aplikasi sistem informasi. Dalam proses ini akan

menghasilkan *design report* yang digunakan untuk proses berikutnya. Dalam *design report* ini terdapat fasilitas untuk mengubah *design report* yang telah dibuat. Pseudocode dari proses desain *report* ini dapat dilihat pada gambar 4.3, gambar 4.4 dan gambar 4.5 di bawah ini.

```

Sub TampilkanDesain()

select * from contentprint where cp_codeaccess ← 'Kode Akses'

if Rs.Recordcount > 0 'Jika ada
    While not rs.EOF
        LoadKomponen rs!Content, lebar, Tinggi, Keterangan
        Rs.movenext
    Wend
end sub

Private Sub LoadKomponen(Content, Lebar, Tinggi, Keterangan)

Membedakan Obyek

Inisialisasi Obyek

Tampilkan Obyek Di Form Desain

End sub

```

Gambar 4.3 Pseudocode untuk Menampilkan Desain Report Berdasarkan Kode Akses

```

Sub TampilkanDesain()

CekKodeAkses
Jika ada
    delete Content print where CP_CodeAccess ← 'KodeAkses'

Urutkan Top Object scr Ascending 'hanya yang terlihat

For i ← 1 to JmlObject
    Tambahkan dalam tabel ContentPrint
next

end sub

```

Gambar 4.4 Pseudocode untuk Menyimpan Desain Report

Public Sub Urut()

I ← 1

For Each object In form

If object.visible Then

Set Sort(i).obj ← *x*

Sort(i).ObjName ← *x.Name*

a ← *x.Top*

Sort(i).Height ← *x.Top*

x ← *Sort(i - 1).Height*

mulai ← *True*

While a < x

If mulai ← *True Then*

Set tmpsort.obj ← *Sort(i - 1).obj*

tmpsort.ObjName ← *Sort(i - 1).ObjName*

tmpsort.Height ← *Sort(i - 1).Height*

Set Sort(i - 1).obj ← *Sort(i).obj*

Sort(i - 1).ObjName ← *Sort(i).ObjName*

Sort(i - 1).Height ← *Sort(i).Height*

Set Sort(i).obj ← *tmpsort.obj*

Sort(i).ObjName ← *tmpsort.ObjName*

Sort(i).Height ← *tmpsort.Height*

r ← *i - 1* 'yang sekarang

s ← *i - 2* 'yang sebelumnya

a ← *Sort(i - 1).Height*

x ← *Sort(i - 2).Height*

mulai ← *False*

Else

Set tmpsort.obj ← *Sort(s).obj*; *tmpsort.ObjName* ← *Sort(s).ObjName*

tmpsort.Height ← *Sort(s).Height*; *Set Sort(s).obj* ← *Sort(r).obj*

Sort(s).ObjName ← *Sort(r).ObjName*

Sort(s).Height ← *Sort(r).Height*; *Set Sort(r).obj* ← *tmpsort.obj*

Sort(r).ObjName ← *tmpsort.ObjName*:

Sort(r).Height ← *tmpsort.Height*

r ← *r - 1*

s ← *s - 1*

a ← *Sort(r).Height*

x ← *Sort(s).Height*

End If

Wend

i ← *i + 1*

End If

Next

JmlObj ← *i - 1*

End Sub



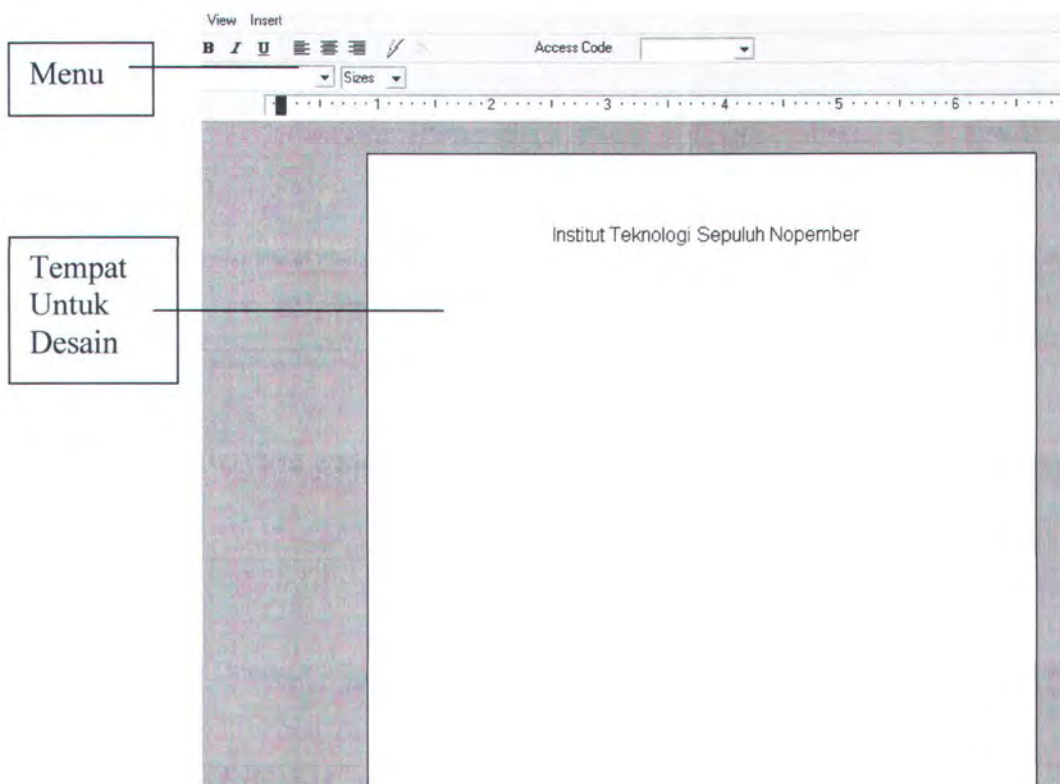
Gambar 4.5 Subrutin untuk Mengurutkan Obyek

4.4 Implementasi Antarmuka

Implementasi antarmuka report designer ini akan membuat tampilan aplikasi sebagai sistem pembatas antara pengguna dan aplikasi. Antarmuka ini akan memudahkan pengguna dalam interaksinya dengan sistem.

Secara umum, implementasi antarmuka ini terdiri atas tujuh bagian utama:

4.4.1 Antarmuka Design Report

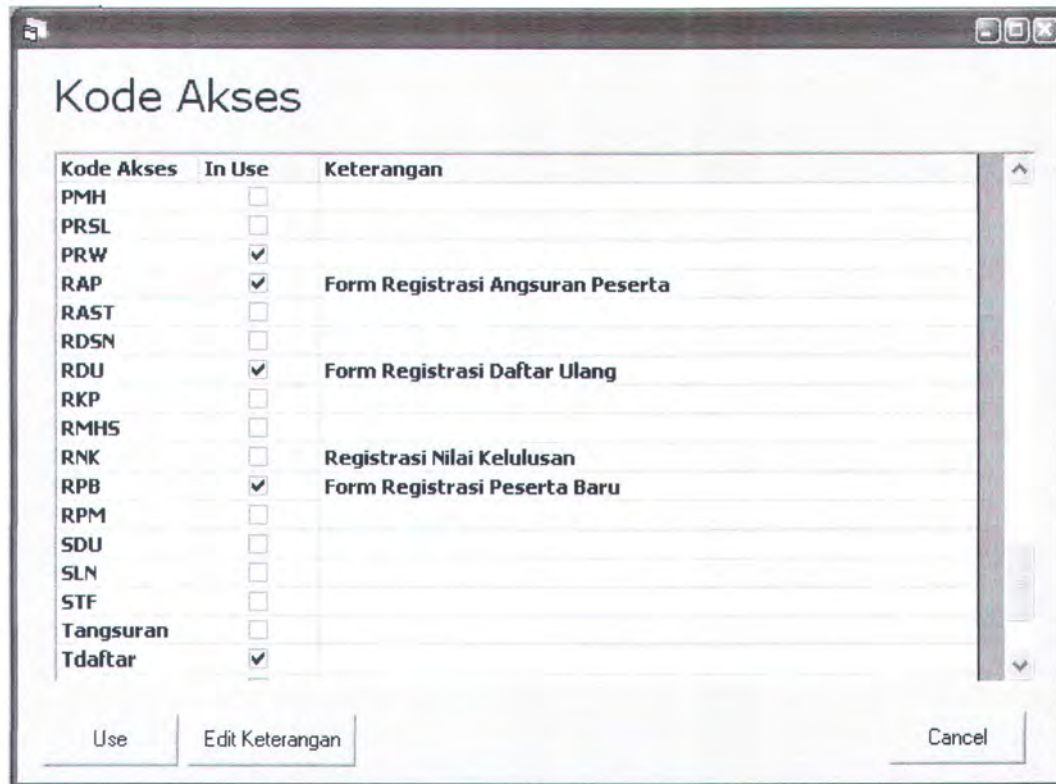


Gambar 4.6 Antarmuka Desain Report

Pada antarmuka ini pihak pengguna dapat mendesain tampilan *report* yang diinginkan oleh pihak pemakai aplikasi sistem informasi. Pada form ini pengguna

dapat secara langsung melakukan *drag and drop* obyek visual basic yang digunakan dalam membuat desain *report*.

1.4.2 Antarmuka Pemilihan Kode Akses



Kode Akses	In Use	Keterangan
PMH	<input type="checkbox"/>	
PRSL	<input type="checkbox"/>	
PRW	<input checked="" type="checkbox"/>	
RAP	<input checked="" type="checkbox"/>	Form Registrasi Angsuran Peserta
RAST	<input type="checkbox"/>	
RDSN	<input type="checkbox"/>	
RDU	<input checked="" type="checkbox"/>	Form Registrasi Daftar Ulang
RKP	<input type="checkbox"/>	
RMHS	<input type="checkbox"/>	
RNK	<input type="checkbox"/>	Registrasi Nilai Kelulusan
RPB	<input checked="" type="checkbox"/>	Form Registrasi Peserta Baru
RPM	<input type="checkbox"/>	
SDU	<input type="checkbox"/>	
SLN	<input type="checkbox"/>	
STF	<input type="checkbox"/>	
Tangsuran	<input type="checkbox"/>	
Tdaftar	<input checked="" type="checkbox"/>	

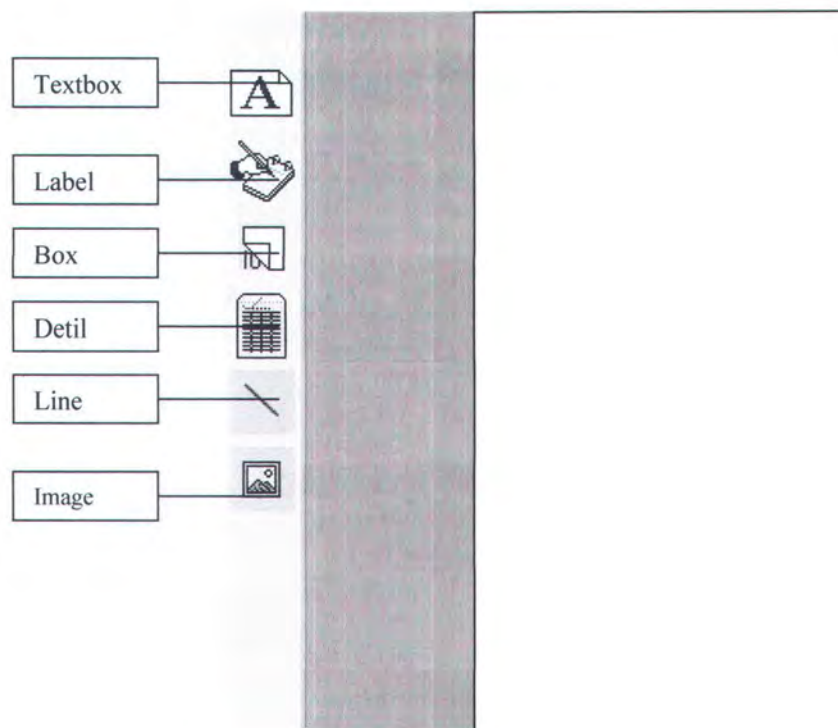
Use Edit Keterangan Cancel

Gambar 4.7 Antarmuka Pemilihan Kode Akses

Antarmuka ini digunakan untuk memilih kode akses yang akan dibuatkan atau di *update* desain *report*-nya. Bila Kolom in use bernilai *true* menandakan bahwa form dengan kode akses tertentu sudah terdapat desain *report*-nya. Sedangkan kolom keterangan berisi tentang keterangan dari kode akses.

Tombol use digunakan untuk menampilkan desain *report* dari kode akses yang telah dipilih. Sedangkan tombol Edit Keterangan digunakan untuk mengubah keterangan dari kode akses yang dipilih.

1.4.3 Antarmuka Obyek Visual Basic yang Digunakan



Gambar 4.8 Antarmuka Obyek VB

Adapun obyek – obyek Visual Basic yang digunakan dalam aplikasi report designer ini adalah :

1. TextBox : Obyek Textbox ini digunakan untuk menampilkan *fields* dari hasil *query* SQL dari kode akses form pada desain *report*.

Properti yang digunakan : Appearance, BorderStyle, DataField, Enabled, Font, ForeColor, Height, ID, Text, Top, Visible, Width.

2. Label : Obyek label ini digunakan untuk membuat tulisan pada desain *report*.

Properti yang digunakan : Appearance, BackStyle, BorderStyle, Caption, Enabled, Font, Height, ID, Left, Top, Visible, Width.

3. Box : Obyek Line ini digunakan untuk menampilkan garis pada desain *report*.

Properti yang digunakan : BorderStyle,ID

4. Detil : Obyek detil ini digunakan untuk menampilkan data detail dari hasil *query*.

Properti yang digunakan : Alignment, Appearance, BorderStyle, *DataField*, Enabled, Font, Height, ID, Text, Top, Visible, Width.

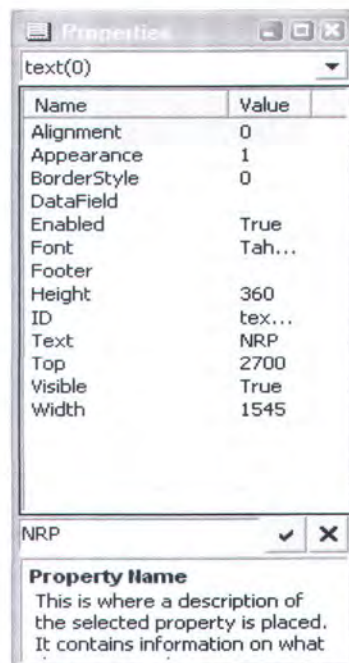
5. Line : Obyek Line ini digunakan untuk menampilkan garis pada desain *report*.

Properti yang digunakan : Appearance, BorderStyle,ID

6. Image : Obyek Image ini digunakan untuk menampilkan gambar pada desain *report*.

Properti yang digunakan : Alignment, Height, ID, Left, Picture, Top, Width

1.4.4 Antarmuka Properti dari Obyek yang Digunakan



Gambar 4.9 Antarmuka property dari VB

Form diatas digunakan sebagai view properti dari obyek visual basic yang dipilih oleh pengguna. Dalam form ini pengguna juga dapat mengganti *value* dari tiap – tiap property dari obyek yang dipilih.

1.4.5 Antarmuka Fungsi

Nama Fungsi	Kegunaan	Full
fr.typeF	Format	<input type="checkbox"/>
fdate	Tanggal	<input checked="" type="checkbox"/>
fterbilang	megubah dari angka ke huruf	<input type="checkbox"/>
fr.IDForm.UserName	nama user	<input checked="" type="checkbox"/>
fTotal	total field	<input type="checkbox"/>
fCounter	No Record	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

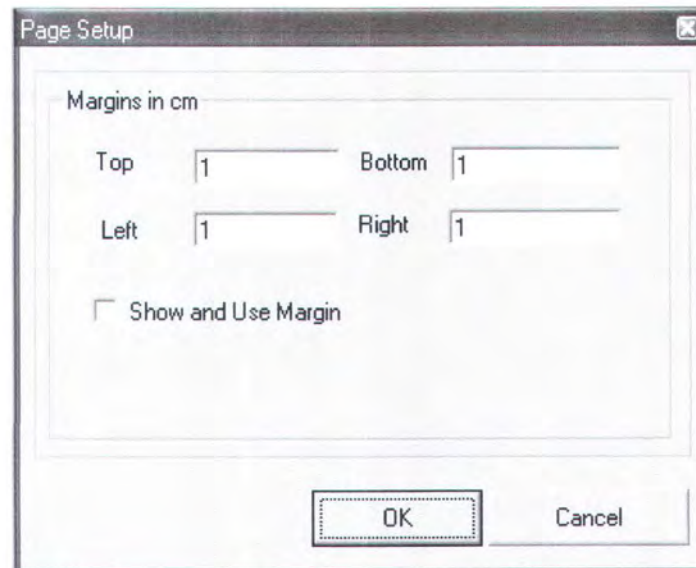
Gunakan Keluar

Gambar 4.10 Antarmuka Fungsi

Form ini digunakan untuk memakai fungsi yang digunakan dalam desain report. Fungsi–fungsi tersebut harus sudah ada dalam proses pembuatan aplikasi sistem informasi. Fungsi–fungsi tersebut dibedakan menurut penggunaanya, bila pada kolom full bernilai benar (*checked*) maka fungsi tersebut tidak dapat diletakkan di dalam *field*. Sebagai contoh fungsi fdate yang berguna untuk menampilkan tanggal. Sedangkan bila pada kolom full bernilai salah (*unchecked*) maka fungsi tersebut harus digunakan bersamaan dengan *field* yang ada dari hasil *query*. Sebagai contoh fungsi fterbilang digunakan untuk mengubah huruf ke angka maka pemakaiannya sebagai berikut :

Fterbilang(namafield)

1.4.6 Antarmuka Pengaturan Kertas dan Margin



Gambar 4.11 Antarmuka Pengaturan Kertas dan Margin

Form ini digunakan untuk menentukan batas margin dari desain *report*. Bila *Show Margin* maka pada form desain akan muncul garis pembatas. Hal ini akan memudahkan pengguna dalam mendesain *report*.

BAB V

UJI COBA

Untuk mengetahui sejauh mana perangkat lunak yang telah dibuat berjalan sesuai yang diharapkan, dilakukan uji coba. Uji coba dilakukan dengan menjalankan beberapa skenario *print*.

5.1 Lingkungan Pengujian Aplikasi

Lingkungan pengujian aplikasi *Report Designer* menggunakan spesifikasi perangkat keras dan perangkat lunak yang dapat dilihat pada tabel 5.1. berikut.

Tabel 5.1 Lingkungan Pengujian Aplikasi

Perangkat Keras	Prosesor : AMD Duron 1,3 GHz
	Memori : 128 MB
	Printer : Canon BJC S200SPx
Perangkat Lunak	Sistem Operasi : Windows XP
	Perangkat Lunak : Microsoft Visual Basic 6.0
	Penyimpanan Desain <i>Report</i> : Microsoft Access 2000
	Penyimpanan Data Aplikasi : SQL Server 2000

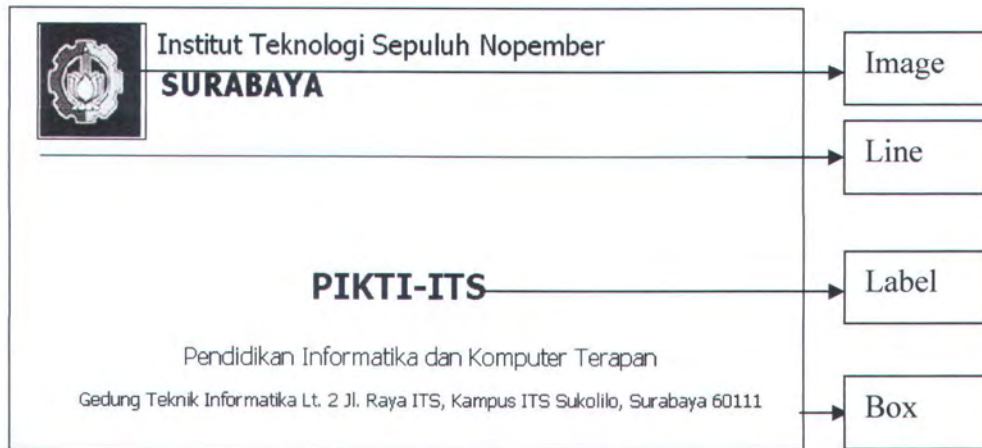
5.2 Pelaksanaan Uji Coba

Terdapat enam skenario uji coba yang akan dilakukan. Pada ujicoba ini aplikasi sistem informasi yang digunakan sebagai output program dari *report designer* adalah aplikasi sistem informasi PIKTI-ITS, sistem informasi akademik

dan sistem informasi gudang. Pertama adalah mendesain dan mencetak tampilan desain *report* tanpa terdapat hubungan dengan *fields* hasil *query* dari kode akses form. Skenario kedua adalah mendesain dan mencetak tampilan desain *report* dimana terdapat hubungan dengan *fields* hasil *query* dari kode akses form. Skenario ketiga adalah mendesain dan mencetak tampilan desain *report* dengan menggunakan fungsi–fungsi yang ada dalam pembuatan aplikasi sistem informasi. Skenario keempat adalah mendesain dan mencetak tampilan desain *report* dengan menampilkan detail. Skenario kelima adalah mendesain dan mencetak tampilan desain *report* dengan menampilkan detail bersebelahan dengan header. Sedangkan Skenario keenam adalah mendesain dan mencetak tampilan desain *report* dengan menampilkan *footer*.

Skenario 1 :

Pihak pembuat aplikasi sistem informasi mendesain *report* sesuai dengan kode akses form yang dipilih tanpa memasukkan *fields* yang ada dari hasil *query* SQL kode akses form dengan memakai semua obyek yang ada pada aplikasi *report designer* kecuali obyek yang berhubungan dengan basis data. Pada skenario ini digunakan sistem informasi PIKTI dengan kode akses RPM yaitu kode akses dari form Pengumuman Kelulusan. Ilustrasi dari proses ini dapat dilihat dari gambar 5.1 berikut. Sedangkan hasil dari pencetakan dapat dilihat pada Gambar 5.2.



Gambar 5.1 Uji Coba Desain Report



Institut Teknologi Sepuluh Nopember
SURABAYA

PIKTI-ITS

Pendidikan Informatika dan Komputer Terapan

Gedung Teknik Informatika Lt. 2 Jl. Raya ITS, Kampus ITS Sukolilo, Surabaya 60111

Gambar 5.2 Hasil Uji Coba Skenario 1

Dari hasil uji coba dapat dilihat bahwa hasil dari pencetakan pada form Pengumuman Kelulusan sesuai dengan desain yang dilakukan pada form tersebut. Hasil pencetakan dari masing-masing obyek sesuai dengan desain yang telah dilakukan.

Skenario 2 :

Pihak pembuat aplikasi sistem informasi mendesain report sesuai dengan kode akses form yang dipilih dengan memasukkan *fields* yang ada dari hasil *query* SQL kode akses form. Pada skenario ini digunakan sistem informasi PIKTI dengan kode akses RDU yaitu kode akses dari form Registrasi Daftar Ulang. Ilustrasi dari proses ini dapat dilihat dari gambar 5.3 berikut. Adapun bentuk *query* yang digunakan dalam kode akses RDU adalah sebagai berikut :

```

"SELECT RPB_ThnAngk, RPB_NoPendaftaran, RPB_NamaPendaftaran,
      RDU_KodeDU = CASE WHEN RDU_KodeDU IS NULL
      THEN 'Belum Daftar Ulang' ELSE RDU_KodeDU END, RDU_Kelas,
      MDU_Uraian, MDU_UangPokok, MDU_TotalBiaya, RPB_ThnAngk,
      RDU_TglDU, RDU_Keterangan, RDU_KodePegawai
FROM REGISTRASIKelulusan INNER JOIN
      REGISTRASIPesertaBaru ON RKM_Kode = RPB_KodeLulus
      LEFT OUTER JOIN REGISTRASIDaftarUlang ON
      RPB_ThnAngk = RDU_ThnAngk
      AND RPB_NoPendaftaran = RDU_NoPendaftaran
      LEFT OUTER JOIN MASTERDaftarUlang ON
      RDU_KodeDU = MDU_KodeDu AND
      MDU_ThnAngk = RDU_ThnAngk LEFT OUTER JOIN Mahasiswa ON
      RDU_NoPendaftaran = MHS_NoPendaftaran
WHERE Upper(RKM_Keterangan) like '%LULUS%' or Upper(RKM_Keterangan) like
      '%CADANGAN%' or Upper(RKM_Keterangan) like '%Wawancara%'
ORDER BY RPB_NoPendaftaran"

```

PIKTI - ITS

Pendidikan Informatika dan Komputer Terapan

Gedung Teknik Informatika Lt 2 Jl. Raya ITS, Kampus ITS Sukolilo, Surabaya 60111

Nomor Reg. : RPB_NoPendaftaran

Atas Nama : RPB_NamaPendaftar

Guna : Pembayaran biaya sebesar MDU_UangPokok

**Gambar 5.3 Uji Coba Desain Report
dengan Field hasil *query***

hasil dari pencetakan dapat dilihat pada gambar 5.4 di halaman berikutnya.



PIKTI - ITS**Pendidikan Informatika dan Komputer Terapan**

Gedung Teknik Informatika Lt 2 Jl. Raya ITS, Kampus ITS Sukolilo, Surabaya 60111

Nomor Reg. : II/96/PIKTI/001

Atas Nama : Eko

Guna : Pembayaran biaya sebesar 500000

**Gambar 5.4 Hasil Pencetakan Uji Coba
Desain Report dengan Field hasil *query***

Dari hasil ujicoba dapat dilihat bahwa hasil dari pencetakan pada form Registrasi Daftar Ulang sesuai dengan desain yang dilakukan pada form tersebut. *Fields* yang tercetak pada hasil pencetakan sesuai dengan *field* yang akan dicetak pada desain report.

Skenario 3 :

Pihak pembuat aplikasi sistem informasi mendesain report sesuai dengan kode akses form yang dipilih dengan memasukkan *fields* yang ada dari hasil *query* SQL kode akses form serta dengan menggunakan fungsi-fungsi. Pada skenario ini digunakan sistem informasi PIKTI dengan kode akses RPB yaitu kode akses dari form Registrasi Pendaftaran Baru. Ilustrasi dari proses ini dapat dilihat dari gambar 5.5 berikut. Adapun bentuk *query* yang digunakan dalam kode akses RPB adalah sebagai berikut :

```

"SELECT RPB_ThnAngk as TahunAngkatan, RPB_NoPendaftaran,
  RPB_NamaPendaftaran, RPB_Tempat, RPB_TglLahir, RPB_Alamat,
  RPB_Telepon, RPB_Agama, RPB_AlamatSBY, RPB_TelpSBY,
  RPB_Tgl_Daftar, RPB_UangDaftar, RPB_Keterangan,
  RPB_Pend_SD, RPB_Pend_SLTP, RPB_Pend_SLTA, RPB_Pend_UNIV,
  RPB_Pend_Lainnya, RPB_Informasi, RPB_KodePegawai
FROM REGISTRASIPesertaBaru "

```



Sudah Terima :	Atas Nama	RPB_NamaPendaftar
Uang	:	fterbilang(RPB_UangDaftar)
Guna	:	Biaya Formulir Pendaftaran Tanggal RPB_Tgl_Daftar
<hr/>		
Terbilang	fr.typeF(RPB_Uang	
		Surabaya , fdate
		fr.IDForm.UserName

**Gambar 5.5 Uji Coba Desain Report
dengan Field hasil *query* dan Fungsi**

Hasil dari pencetakan dapat dilihat pada Gambar 5.6 di halaman berikutnya.

Sudah Terima : Atas Nama Pratiwanggana Sigit
Uang : Tujuh puluh Lima ribu
Guna : Biaya Formulir Pendaftaran Tanggal 06/03/2001

Terbilang 75.000

Surabaya , 12/10/2005

Dwi Sunaryono

**Gambar 5.6 Hasil Pencetakan Uji Coba
Desain Report dengan Field hasil *query*
dan Fungsi**

Dari hasil ujicoba dapat dilihat bahwa hasil dari pencetakan pada form Registrasi Pendaftaran Baru sesuai dengan desain yang dilakukan pada form tersebut. *Fields* yang tercetak pada hasil pencetakan sesuai dengan *field* yang akan dicetak pada desain report. Selain itu fungsi yang tercetak juga sesuai dengan fungsi yang digunakan pada desain report.

Skenario 4 :

Pihak pembuat aplikasi sistem informasi mendesain report sesuai dengan kode akses form yang dipilih dengan memasukkan *fields* yang ada dari hasil *query* SQL kode akses form serta dengan serta dengan menampilkan data detail. Pada skenario ini digunakan sistem informasi PIKTI dengan kode akses RAP yaitu kode akses dari form Registrasi Angsuran Peserta. Ilustrasi dari proses ini dapat dilihat dari gambar 5.7 berikut. Adapun bentuk *query* yang digunakan dalam kode akses RAP adalah sebagai berikut :

```

"SHAPE {SELECT RPB_ThnAngk, RPB_NoPendaftaran,
  RPB_NamaPendaftaran, MDU_Uraian,
  MDU_TotalBiaya AS BiayaSeluruhnya,
  MDU_UangPokok + ISNULL(SUM(MDC_UangCicilan), 0)
  AS TotalTelahBayar,
  MDU_TotalBiaya - MDU_UangPokok - ISNULL(SUM(MDC_UangCicilan), 0)
  AS Kekurangan, MDC_UangCicilan FROM REGISTRASIDaftarUlang
  INNER JOIN REGISTRASIPesertaBaru ON RDU_ThnAngk = RPB_ThnAngk AND
  RDU_NoPendaftaran = RPB_NoPendaftaran LEFT OUTER JOIN
  MASTERDaftarUlang ON RDU_ThnAngk = MDU_ThnAngk AND RDU_KodeDU
  = MDU_KodeDu
  LEFT OUTER JOIN REGISTRASIAngsuranPeserta ON
  RDU_ThnAngk = RAP_ThnAngk AND RDU_NoPendaftaran = RAP_NoPendaftaran
  LEFT OUTER JOIN MASTERDaftarCicilan ON MDU_ThnAngk = MDC_ThnAngk
  AND MDU_KodeCicilan = MDC_KodeCicilan AND RAP_ThnAngk = DC_ThnAngk
  AND RAP_KodeCicilan = MDC_KodeCicilan
  GROUP BY RDU_NoPendaftaran, MDU_TotalBiaya, MDU_UangPokok,
  RPB_ThnAngk, RPB_NoPendaftaran, RPB_NamaPendaftaran, MDU_Uraian,
  MDC_UangCicilan} AS ParentCMD APPEND
  ({SELECT RAP_ThnAngk, RAP_NoPendaftaran, RAP_KodeCicilan, AP_TglCicilan,
  RAP_CicilanKe, RAP_TglTransferBank, RAP_Keterangan,
  RAP_KodePegawai FROM REGISTRASIAngsuranPeserta Order by RAP_TglCicilan
  }
  AS ChildCMD RELATE RPB_NoPendaftaran TO RAP_NoPendaftaran) AS ChildCMD"

```

Header

Nama	RPB_NamaPendaftar
------	-------------------

Biaya Seluruhnya	BiayaSeluruhnya
------------------	-----------------

Detil

No Pendaftaran	Tgl Cicilan	Cicilan Ke-	Tgl Transfer Bank
RAP_NoPende	RAP_TglCicilan	RAP_I	RAP_TglTrans

**Gambar 5.7 Uji Coba Desain Report
dengan Field hasil *query* dan data detil**

Hasil dari pencetakan dapat dilihat pada Gambar 5.8 di halaman berikutnya.

Registrasi Angsuran Peserta

Header

Nama	Eko
Biaya Seluruhnya	2000000

Detil

No Pendaftaran	Tgl Cicilan	Cicilan Ke-	Tgl Transfer Bank
II/96/PIKTI/001	07/08/1996	1	05/05/2003
II/96/PIKTI/001	30/09/1996	3	30/09/1996
II/96/PIKTI/001	30/09/1996	2	30/09/1996

**Gambar 5.8 Hasil Pencetakan Uji Coba
Desain Report dengan Field hasil *query*
dan data detil**

Dari hasil ujicoba dapat dilihat bahwa hasil dari pencetakan pada form Registrasi Angsuran Peserta sesuai dengan desain yang dilakukan pada form tersebut. *Fields* yang tercetak pada hasil pencetakan sesuai dengan *field* yang akan dicetak pada desain report.

Skenario 5 :

Pihak pembuat aplikasi sistem informasi mendesain report sesuai dengan kode akses form yang dipilih dengan memasukkan *fields* yang ada dari hasil *query* SQL kode akses form serta dengan serta dengan menampilkan data detail bersebelahan dengan header. Pada skenario ini digunakan sistem informasi Akademik dengan kode akses JUM yaitu kode akses dari form Jadwal UAS Mahasiswa. Adapun bentuk *query* yang digunakan dalam kode akses JUM adalah sebagai berikut :

```
"SHAPE { select * from mahasiswaku} AS ParentCMD APPEND
({select * from vJadwalUAS} AS ChildCMD RELATE NRP TO NRP)
AS ChildCMD"
```

Ilustrasi dari proses ini dapat dilihat dari gambar 5.9 berikut.

JADWAL UAS Th. AJARAN 2003 - 2004 SEMESTER GENAP		Teknik Informatika ITS					
DATA MAHASISWA		DAFTAR MATA KULIAH					
NRP	NRP	Tanggal	Hari	Jam	Kode	Nama Matakuliah	Ruang
NAMA	Nama						
ALAMAT	Alamat						

Gambar 5.9 Uji Coba Desain Report dengan Field hasil *query* dan data detil bersebelahan dengan header

Hasil dari pencetakan dapat dilihat pada Gambar 5.10.

JADWAL UAS Th. AJARAN 2003-2004 SEMESTER GENAP

Teknik Informatika ITS

Data Mahasiswa

Daftar Mata Kuliah

		Tanggal	Hari	Jam	Kode	Mata Kuliah	Ruang
NRP	5199100013	22/06/2004	Selasa	11.00	IF1234	SPK	TC102
		24/06/2004	Kamis	10.00	IF1345	SI	TC101
		21/06/2004	Senin	10.00	IF1640	PSI	TC106
		22/06/2004	Selasa	08.00	IF1670	SBP	TC104
		23/06/2004	Rabu	11.00	IF1671	PSSI	TC101
Nama		Aryo					
Alamat		Surabaya					

**Gambar 5.10 Hasil Pencetakan Uji Coba
Desain Report dengan Field hasil *query* dan
data detil bersebelahan dengan header**

Data Distribusi Barang

ID Gudang ID_Gudang

Nama Gudang Nama_Gudang

Alamat Alamat_Gudang

No	Kode	Jumlah	Harga
fCo	KodeBara	Jml	HargaBara
Total		fTotal/	fTotal (Ha

**Gambar 5.11 Uji Coba Desain Report dengan Field
hasil *query* dan data detil beserta *footer***

Hasil dari pencetakan dapat dilihat pada Gambar 5.12 di halaman berikutnya

Data Distribusi Barang

ID Gudang G002

Nama Gudang Gudang B

Alamat Surabaya

No	Kode	Jumlah	Harga
1	G002	5	1000
2	G002	3	1000000
3	G002	5	1300000
4	G002	7	1700000
5	G002	4	1100000
6	G002	5	1350000
7	G002	8	200000
Total		37	6651000

Gambar 5.12 Hasil Uji Coba Desain Report dengan Field hasil *query* dan data detil beserta *footer*

Dari hasil ujicoba dapat dilihat bahwa hasil dari pencetakan pada form Distribusi Barang sesuai dengan desain yang dilakukan pada form tersebut. *Fields* dan *footer* yang tercetak pada hasil pencetakan sesuai dengan *field* dan *footer* yang akan dicetak pada desain report.



BAB VI

PENUTUP

Bab terakhir dalam penulisan Tugas Akhir ini berisi kesimpulan dan kemungkinan pengembangan lebih lanjut dari aplikasi *report designer* yang telah dibuat.

6.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari pelaksanaan tugas akhir ini adalah sebagai berikut:

- Aplikasi *report designer* ini merupakan alat bantu pembuatan laporan bagi aplikasi sistem informasi dengan menggunakan metode Hierarchical Task Analysis (HTA).
- Dengan menggunakan aplikasi *report designer* ini tidak perlu dilakukan rekompilasi dikarenakan proses bisnis program sudah tersimpan dalam basis data.
- Dengan menggunakan aplikasi *report designer* ini tidak perlu file khusus *report* bagi aplikasi sistem informasi yang dibangun dikarenakan desain *report* sudah menjadi satu dengan basis data HTA.
- Aplikasi *report designer* dapat melakukan pencetakan gambar, garis, kotak serta *fields* dari hasil *query*, dan penggunaan fungsi. Selain itu pencetakan dengan data detil juga dapat dilakukan.

6.2 Kemungkinan Pengembangan

Hal-hal yang masih dimungkinkan untuk dikembangkan pada aplikasi *report designer* ini adalah:

- Penambahan fitur-fitur desain *report* seperti *support store procedure*, dapat membuat fungsi-fungsi tambahan selain fungsi-fungsi built-in, dan lain sebagainya.
- Aplikasi *report designer* dapat dikembangkan menjadi aplikasi berbasis web sehingga dapat mendukung pembuatan aplikasi sistem informasi berbasis web.

DAFTAR PUSTAKA

- [1] Dix, Alan, Janet Finlay dan Russel Beale : *Human – Computer Interaction 3rd edition*. Prentice Hall (2004)
- [2] Guy Eddon & Henry Eddon, *InProgramming Components With Microsoft Visual Basic 6.0*, Microfot Press, 1998
- [3] Michael C.Reingruber dan William W. Gregory, *Data Modeling Handbook*, Canada,1994
- [4] Smith dan Ragan, *Some Basic Differences between A Procedural Task Analysis and A Hierarchical Task Analysis*. Tersedia : http://classweb.gmu.edu/ndabbagh/Resources/hierarchy_vs_procedural.htm, 1994
- [5] Stanley,Tina : *Hierarchical Task Analysis*. Tersedia : http://classweb.gmu.edu/ndabbagh/Resources/hierarchical_analysis.htm , 1999